

AD-A275 402



CMU/SEI-93-SR-19

2



Carnegie-Mellon University
Software Engineering Institute

Draft SEI Program Plans: 1994-1998

August 1993

DTIC
ELECTE
FEB 03 1994
S E D

Approved for public release
Distribution unlimited

98 12 18 06 6

93-30450



CMU/SEI-93-SR-19

Draft SEI Program Plans: 1994-1998

DTIC QUALITY INSPECTED 6

Approved for public release.
Distribution unlimited.

| | |
|--------------------|---|
| Accession For | |
| NTIS | CRA&I <input checked="" type="checkbox"/> |
| DTIC | TAB <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By | |
| Distribution / | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

August 1993

This technical report was prepared for the

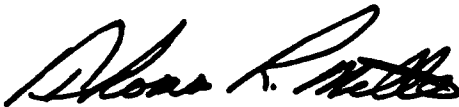
SEI Joint Program Office
ESC/ENS
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.

This report was funded by the U.S. Department of Defense.

Copyright © 1993 by Carnegie Mellon University.

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Service. For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

Copies of this document are also available from Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212, Telephone: (412) 321-2992 or 1-800-685-6510, Fax: (412) 321-2994.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.



Carnegie Mellon University
Software Engineering Institute

August 17, 1993

Colleagues:

I'm pleased that you've chosen to read the *Draft SEI Program Plans: 1994-1998* (also known as the SEI 1&5 Year Plan). This document presents the SEI strategy and one-year implementation plan for calendar year 1994, together with the SEI five-year program plans.

Every year, we prepare a similar document to submit to our sponsor as a contract deliverable. This year and for the first time, we decided to make this version available for public release. This document provides an insight into the direction of the SEI and our planned activities, products, and services for the future.

This document is a draft plan. Its execution depends primarily on resource allocations. The planning starts long before the Congress completes its budget authorization and appropriation. Historically, circumstances such as changing customer needs and changing resource allocations have made it necessary to change our plans. For example, when we wrote this document, we planned a Reengineering Workshop for the third quarter of 1994. Because of the strong interest in this area, we've now rescheduled this workshop for October 1993.

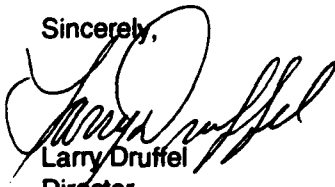
In reading this document, please consider opportunities in which you can work with us. As discussed in Section 4.4, the SEI has developed a range of relationships that provide mutual benefit to us and our customers in industry, government, and academia. These relationships include the subscriber program, the resident affiliate program, distribution/transition partnerships, advisory boards and working groups, Software Process Improvement Network (SPIN) organizations, and technical and strategic partnerships. I invite you to investigate which opportunities are right for you and your organization.

To contact us at the SEI, write or call:

Customer Relations
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Phone: (412) 268-5800
FAX: (412) 268-5758
Internet: customer-relations@sei.cmu.edu

We look forward to working with you toward our common goal of improving the practice of software engineering.

Sincerely,



Larry Druffel
Director

ld:wkm

Table of Contents

| | |
|---|-----------|
| Introduction | 1 |
| | |
| 1 Strategic Context | 3 |
| 1.1 Charter | 3 |
| 1.2 Mission, Vision, and Strategy | 4 |
| 1.3 Orientation | 5 |
| 1.4 Customers | 6 |
| 1.4.1 Sponsors and Partners | 6 |
| 1.4.2 Acquisition, Development, and Post Deployment | 7 |
| 1.4.3 Managers, Practitioners, and Educators | 7 |
| | |
| 2 Strategic Overview | 11 |
| 2.1 Situation Analysis | 12 |
| 2.1.1 Shrinking DoD Budgets and Downsizing of the Military | 12 |
| 2.1.2 The Changing Role of the Military | 12 |
| 2.1.3 The Increasing Concern for the Domestic Infrastructure | 13 |
| 2.1.4 The Increase in Global Competitiveness | 13 |
| 2.1.5 Effect of Major Trends on Software | 14 |
| 2.2 Strategy for Improving Software Engineering Practice | 15 |
| 2.2.1 Software Process Definition, Modeling, and Measurement | 17 |
| 2.2.2 Methods and Tools for Disciplined Engineering of Software Systems | 18 |
| 2.2.3 Software Technology Transition | 18 |

| | |
|--|------------|
| 2.3 Planning Considerations | 19 |
| 2.4 Conclusion | 20 |
| 3 Technical Focus Areas | 23 |
| 3.1 Overview | 23 |
| 3.1.1 Maturing the Profession | 23 |
| 3.1.2 Maturing the Process | 23 |
| 3.1.3 Maturing the Technology | 25 |
| 3.1.4 Technical Focus Areas | 25 |
| 3.1.5 Measures of Success | 28 |
| 3.1.6 Effects of Unpredictable Funding | 28 |
| 3.2 Process | 29 |
| 3.2.1 Five-Year Plan | 30 |
| 3.2.2 One-Year Plan | 41 |
| 3.3 Risk | 45 |
| 3.3.1 Five-Year Plan | 47 |
| 3.3.2 One-Year Plan | 57 |
| 3.4 Methods and Tools | 62 |
| 3.4.1 Five-Year Plan | 64 |
| 3.4.2 One-Year Plan | 79 |
| 3.5 Real-Time Distributed Systems | 85 |
| 3.5.1 Five-Year Plan | 88 |
| 3.5.2 One-Year Plan | 96 |
| 4 Technology Transition | 103 |
| 4.1 Technology Transition Models | 104 |
| 4.1.1 Investigations into Technology Transition Models | 108 |

| | |
|--|-----|
| 4.2 Products and Services | 113 |
| 4.2.1 Manager Products | 114 |
| 4.2.2 Practitioner Products | 115 |
| 4.2.3 Educator Products | 116 |
| 4.2.4 Summary of SEI Product Plans | 116 |
| 4.3 Transition Partners | 124 |
| 4.4 Customer Involvement | 126 |
| 4.4.1 Customer Inquiry/Response | 126 |
| 4.4.2 Subscriber Program | 126 |
| 4.4.3 Resident Affiliate Program | 127 |
| 4.4.4 Advisory Boards/Working Groups | 128 |
| 4.4.5 Software Process Improvement Network | 128 |
| 4.4.6 Technical and Strategic Partnership Programs | 129 |
| 4.4.7 SEI Software Engineering Symposium | 130 |
| 4.5 Education in Technology Transition | 132 |
| 4.5.1 Course Development and Delivery | 132 |
| 4.5.2 Curriculum Influence | 137 |
| 4.6 Services in Technology Transition | 140 |
| 4.6.1 Five-Year Plan | 140 |
| 4.6.2 One-Year Plan | 142 |

| | |
|---|------------|
| 5 Relationship of Focus Areas to Software Topics | 145 |
| 5.1 Reuse | 145 |
| 5.2 Reengineering | 146 |
| 5.3 Testing | 146 |
| 5.4 Software Maintenance | 147 |
| 5.5 Simulation | 147 |
| 5.6 Open Systems | 147 |
| Appendix A Currently Available SEI Products and Services | 149 |
| Appendix B Resident Affiliates | 153 |
| Appendix C Transition Partners | 155 |
| References | 161 |
| List of Acronyms | 163 |

List of Figures

| | | |
|---------------------|--|-----------|
| Figure 2-1: | Strategic Framework and Core Competencies for Improving Software Practice | 17 |
| Figure 3-1: | Benefits in Moving from Level 1 to Level 3 | 25 |
| Figure 3-2: | Strategic Framework, Core Competencies, and Focus Areas | 27 |
| Figure 3-3: | Maturity Level of Assessed Sites 1987-89 vs 1990-92 | 29 |
| Figure 3-4: | Five-Year Technical Product Roadmap for the Process Focus Area | 40 |
| Figure 3-5: | SEPG Workshop Registrants | 42 |
| Figure 3-6: | SCE Activity | 43 |
| Figure 3-7: | Risks Within a System Context | 45 |
| Figure 3-8: | Balancing Opportunities and Risk | 46 |
| Figure 3-9: | Risk Management Paradigm | 49 |
| Figure 3-10: | Risk Product Areas | 51 |
| Figure 3-11: | Five-Year Technical Product Roadmap for the Risk Focus Area | 56 |
| Figure 3-12: | Risk Data Repository | 59 |
| Figure 3-13: | Model-Based Software Engineering Practice | 63 |
| Figure 3-14: | Coordinated MBSE Thrusts | 66 |
| Figure 3-15: | MBSE Process | 69 |
| Figure 3-16: | Tool Integration Technology Streams | 71 |
| Figure 3-17: | Common View of Reengineering | 72 |
| Figure 3-18: | System Understanding | 73 |
| Figure 3-19: | Foundation Work Leading to Model-Based Software Engineering | 74 |

| | | |
|---------------------|---|------------|
| Figure 3-20: | Five-Year Technical Product Roadmap for the Methods and Tools Focus Area | 78 |
| Figure 3-21: | Mapping of Tasks to Resources | 87 |
| Figure 3-22: | Technologies for Dependable Systems | 92 |
| Figure 3-23: | Organization of the Handbook Series | 93 |
| Figure 3-24: | Five-Year Technical Product Roadmap for the Real-Time Distributed Systems Focus Area | 95 |
| Figure 3-25: | System Relationships | 96 |
| Figure 3-26: | A Structural Model | 98 |
| Figure 3-27: | Science and Technology Maturity | 99 |
| Figure 3-28: | Interaction Between Systems Requirements | 100 |
| Figure 4-1: | Conceptual Framework for Technology Transition | 105 |
| Figure 4-2: | Commitment to Technological Change | 106 |
| Figure 4-3: | Effect of Software Process Improvement Networks | 130 |

Introduction

This document presents the Software Engineering Institute (SEI) strategy and one-year implementation plan for calendar year (CY) 1994, together with the SEI five-year program plan.

In Chapter 1, we set the strategic context by discussing the SEI charter, mission, vision, strategy, orientation, and customers—managers, practitioners, and educators.

In Chapter 2, we describe the factors that determine SEI plans and set the context for their implementation in support of the SEI mission: *to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software.*

In Chapter 3, we describe the SEI technical program. The goal of the SEI technical program is to improve software engineering practice by:

- Maturing the skills of practitioners who develop and maintain software and of managers who organize and lead those activities (Maturing the Profession).
- Maturing the organizational and managerial processes through which practitioners develop and maintain software (Maturing the Process).
- Maturing the technology used by practitioners to develop and maintain software (Maturing the Technology).

These three strategic activities—maturing the profession, maturing the process, and maturing the technology, combined with the SEI core competence in software technology transition—comprise the strategic framework that guides SEI technical efforts.

We have organized our technical program into clusters of related activities called focus areas to identify and transition those technologies that we believe will mature the profession, the process, and the technology. The four focus areas are: process, risk, methods and tools, and real-time systems. Work conducted in the focus areas also maintains the SEI core competencies in software process definition, modeling, and measurement; methods and tools for disciplined engineering of software systems, and software technology transition.

1. Through our focus on **process**, we are presently concerned with the maturity of the organizational and managerial processes employed by software development organizations. The SEI seeks to define, model, measure, and improve the maturity of these processes. The expectation is that doing so will improve the organizational performance in developing software.
2. Our technical focus on **risk** provides a systematic and structured process, supported by methods and tools, for identifying and analyzing the technical uncertainties encountered in a specific software development. Having such a process will significantly enhance the probability of success of a program by allowing risk-resolving steps to be taken before problems occur.

3. Through our next focus area, we are developing **methods and tools** that operate on or within the software for defining, analyzing, and evaluating domain models and architectures for the disciplined engineering of software systems.
4. Our focus in **real-time systems** is motivated by the need for systems that must satisfy critical real-time constraints. Our goal has been to provide methods and tools for creating such systems that can guarantee that timing constraints are met. Our goal is now broadened to address dual-use (defense and civilian) applications where quality attributes such as timeliness, reliability, safety, and interoperability are important.

Each SEI focus area is presented in depth, including the problems being addressed, the approach to solving them, and potential products that can result from their solution. Also included in this chapter are sections providing context with last year's *SEI Program Plans*, including 1993 activities that provide a foundation for 1994 activities, and significant changes in direction; and sections that relate technical objectives and plans (TO&P) funded activities with core funded activities.

In Chapter 4, we give a complementary view of the SEI program, from the perspective of technology transition. The SEI mission requires a technology transition strategy that gives us leverage in meeting the needs of our customers. We have learned that development and delivery of products and services offer the most effective way to accomplish transition. We describe the models on which our transition strategy is based, and we briefly describe the types of SEI products and services available from the SEI and the customers who use them: managers, practitioners, and educators. This chapter includes a list of products planned for 1994 and potential products to be developed by 1999.

Delivering these products and services to the software engineering community is a challenge for a small organization like the SEI. To gain leverage, we work with transition partners, who assist us in tailoring and delivering our products. We also involve our customers in several ways, providing timely information on SEI activities and receiving their input on our products and services. In Chapter 4, we also discuss the significant role of education and services in technology transition.

In Chapter 5, we review the SEI program from a programmatic point of view. We summarize the overlap of our focus areas with key software topics, such as reuse, reengineering, testing, software maintenance, simulation, and open systems. Each of these topics is discussed in the context of the SEI activities to which they relate.

1 Strategic Context

The purpose of this chapter is to establish the strategic context for the five-year plan and one-year implementation.

The Software Engineering Institute (SEI) was established in 1984 by Congress as a federally funded research and development center (FFRDC) with a broad charter to address software engineering transition. The SEI is funded by the Advanced Research Projects Agency (ARPA) through a contract with the Air Force Materiel Command/Electronic Systems Center (AFMC/ESC). These relationships establish organizational, funding, and reporting structures; they provide a comparative advantage and natural focus for selecting customers and activities.

The SEI is an integral component of Carnegie Mellon University (CMU), which carries with it the responsibility to maintain equivalent quality in staff and in the conduct of activities. As a member of the CMU community and as an ARPA-funded organization, the SEI is an active participant in the larger software research community.

1.1 Charter

The SEI charter is to:

- Bring the ablest professional minds and the most effective technology to bear on the rapid improvement of the quality of operational software in systems that depend on software.
- Accelerate the reduction to practice of modern software engineering techniques and methods.
- Promulgate the use of modern techniques and methods throughout the defense community.
- Establish standards of excellence for software engineering practice.

Twenty-five percent of SEI core activity is chartered for research and education. The remaining seventy-five percent is for technology transition. In addition, the SEI may receive funding from federal agencies other than ARPA for specified work consistent with the charter.

1.2 Mission, Vision, and Strategy

Software is an enormous opportunity, offering cost-effective flexibility for military as well as commercial systems. Historically, our customers have experienced significant difficulties in acquiring, deploying, and maintaining large-scale software systems. Software often does not meet expectations, is delivered late and over budget, and is difficult to change to meet evolving needs. We believe that these problems can be avoided by bringing an engineering discipline to the way software is created. The current state of the practice is far behind the state of the art.

Our **mission** is to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software.

We want our customers to be capable of applying a mature software engineering discipline to produce high quality software that meets their expectations, at a competitive price and on predictable schedules. Therefore, we are committed to the evolution of software engineering from an ad-hoc, labor-intensive activity to a managed, technology-supported engineering discipline.

We **envision** ourselves as an organization opening gateways to improved software engineering practice. The analogy is that of computer networks in which gateways offer access to information and capability. We see our role as improving the practice by establishing human and technology connections that will allow improved practices to spread throughout the industry.

Our intent is to identify and transition to customers, through transition products and services, those processes, methods, and tools that will help them make lasting improvements to their overall software engineering capabilities.

Our **strategy** for implementing this intent is to:

- Understand, model, and assess selected software practices required to define, develop, and maintain complex systems.
- Select software engineering technical areas of strategic importance to U.S. leadership in software, with emphasis on defense systems.
- Develop an objective body of expertise at the SEI in these areas that is unavailable elsewhere.
- Offer a cohesive set of products and services to help managers, practitioners, and educators make lasting improvements in their software engineering practices.

In applying this strategy, we will focus our technical activities in software engineering technology areas of critical importance to our customers. We will continue to address other important software engineering issues, but will not seek to establish a leadership position in those other areas. To accomplish a leadership position in an area of focus requires at least 25-30 people with appropriate expertise, and an additional cadre of specialized support. With our size constraints, we cannot expect to focus in more than five areas. A minimum of four areas appears necessary to have the broad impact envisioned in the charter.

In each focus area, we research, evaluate, mature, and demonstrate technology solutions in a realistic environment. Demonstrations are planned so that (1) the SEI can determine whether a product or service should be developed, (2) risk of adoption is reduced in the eyes of potential customers, and (3) the costs and benefits of adoption are measured to support an acceptable return on investment (ROI) to SEI customers.

Also in each focus area, we identify (1) the customer base; (2) customer strategic intent, needs, and requirements; (3) our vision, goals, and objectives; and (4) the specific products we will develop to achieve those goals and objectives.

Our products and services include courses, events, publications, prototype software, videotapes, and guidance and advice in the use of our products. These products and services help the software community improve its management practices, technical practices, and the capabilities of its personnel. For more details, see Chapter 4.

1.3 Orientation

As a technology organization, the SEI promotes software engineering and supporting technology. Technology is our strength, and we must be technology driven. However, we do not promote technology for its own sake; we are also needs driven. A need can result from an encountered problem, an opportunity enabled by innovation, or anticipation of future problems or technological advances. We help organizations understand the root causes of their software engineering problems as needs. Four considerations influence which problems we work on:

1. The mission to advance the state of the practice of software engineering requires the SEI to have a broad impact by concentrating on those problems that are pervasive.
2. The SEI is in a trusted position that demands objectivity. Organizations expect the SEI to exert independent technical judgment and influence based on a broad and deep understanding of the field, and to understand and provide solutions to the root causes of problems, not simply to eliminate a symptom.
3. The SEI is a relatively small organization. More needs and problems exist than we can address, and there is more work to be done than we can expect to accomplish. We must be selective in choosing problems that are strategically important and have high-leverage potential, understand where outside expertise is available, and work within our abilities.
4. The SEI, by contract, is not permitted to compete in markets predictably and properly satisfied by commercial enterprise.

We are committed to be a needs-driven organization in this sense and have made this orientation an explicit part of our business. We will pursue technologies that offer solutions to real needs. To have a broad impact, we will provide solutions in the form of products and services that help organizations help themselves.

1.4 Customers

Customers are beneficiaries of SEI products and services. The SEI has many customers in the Department of Defense (DoD), a few in other federal agencies, and, implicitly, many in industry and academia. The latter develop much of the DoD software and train software practitioners. To better serve our government customers, we have identified two special categories of customers (**sponsors** and **partners**) with whom we collaborate in the development, maturation, and initial transition of needed products and services.

1.4.1 Sponsors and Partners

Sponsors invest funds in the development of capability. For instance, a sponsor might fund the SEI to investigate a certain technology. Sponsorship may be tied to the condition that the sponsor be the initial customer of a resulting product. A specific SEI activity could have multiple sponsors.

The DoD, through ARPA, is a major sponsor that invests funds in the SEI base (core funding). This enables the SEI to understand needs, evaluate technology, propose and test solutions, and then to develop and demonstrate products and services for our customers. These base funds also enable the SEI to develop and maintain relationships with the supporting software infrastructure in the United States.

A significant portion of the SEI's total resources is received through technical objectives and plans (TO&P) funding. Whereas core funding enables the institute to investigate emerging ideas and technologies, TO&P agreements provide the means for the SEI to put promising results into practice for specific customers. This type of interaction establishes a near-term conduit for SEI products and services to flow into the software community, and it permits the SEI to maintain insight into the nature of software practice. Through TO&P agreements, the SEI works in the field to promote and verify improved practices in conjunction with the sponsor and to gather data that will further inform future efforts of a like nature.

Partners collaborate in the development, demonstration, or transition of SEI products or services. They may benefit directly or indirectly by the partnership. They also assume some risk. They contribute to the success of a specific product by providing expertise, perspective, credibility, and/or delivery capability. Organizations that send resident affiliates (that is, individuals on long-term assignment at the SEI from their home institutions) are, by definition, partners.

Partners provide us with insight into problems, assist with testing SEI products, or offer a context for demonstrating solutions. The primary consideration in matching our capabilities to specific partners' needs is the credibility partners bring to the test or demonstration. They should

bring specialized expertise to the SEI or be representative of a class of potential customers. They will be selected based on their contribution to the success of an activity, their relative importance to an SEI sponsor, or their contribution to the sponsor. In addition, commercial vendors may provide leverage for SEI products by becoming transition partners who service broader markets than the SEI would be able to serve.

1.4.2 Acquisition, Development, and Post Deployment

Our DoD customers focus on three distinct phases of the software life cycle: (1) acquisition, (2) development, and (3) post-deployment support. Each phase generates somewhat different software engineering concerns.

Acquisition is the phase in which requirements are defined and contracts are let for software development to meet these requirements. Concerns of acquisition organizations include policy, standards, requirements definition, cost and schedule estimation, contract and risk management, reengineering, reuse, training, and testing.

Development is the phase in which software is created that satisfies the requirements of the contracts resulting from the acquisition phase. The concerns of development organizations include requirements, specification, design, coding, integration, testing, risk management, installation, training, and project management.

Post-deployment is the phase that addresses the support of the software after the system is fielded (operational). The principal concerns of post-deployment software support (PDSS) organizations are reliability, maintainability, reengineering, and the costs associated with these. Maintenance of the software within a system addresses two primary aspects: the correction of latent defects (commonly referred to as "bugs"), and planned evolutionary enhancements that improve system functionality. Enhancements are accomplished by the same process that is conducted in the development phase.

In serving our customers, we have identified that the needs of managers, practitioners, and educators differ, and we have tailored our product offerings accordingly. In the following section, we describe our understanding of those needs from the perspective of our principal customers. We recognize that managers, practitioners, and educators in other sectors have similarly defined needs.

1.4.3 Managers, Practitioners, and Educators

Managers concentrate on system acquisition that encompasses all phases of the life cycle: research, development, production, and operation. Acquisition is performed by an organization representing the end user of a software-intensive system.

Each military service has program executive officers (PEO) who serve as system materiel developers responsible for acquisition of the system. PEO organizations are co-located with supporting functional commands and have small staffs. Mission accomplishment is through the use of the matrix concept, where functional services and expertise are supplied by supporting

functional commands, e.g., life-cycle software engineering centers. System development is generally performed for the PEO by industry through a contract. The PEO is responsible for the management of the system development through a statement of work that is specified within the contract.

Generally, the responsibility for the systems software technical management of a life-cycle software support (LCSS) center is assigned to a program manager (PM) for ensuring software product quality. In this principal role, the center must exercise oversight for the PM in working with the contractor for all software quality assurance that eventually leads to final acceptance testing. Contractor management is responsible to the PM for complying with the specifications within the contract and responsible for managing the technical development of the system in the most cost-effective way while ensuring high-quality software.

Although we support PEOs who are in the acquisition business and we must therefore develop appropriate methodology (e.g., software capability evaluation (SCE)), we ourselves are not in the acquisition business. Our chartered role of objective broker specifically precludes our participation in source selection activities or those activities involving government/contractor negotiation. However, we can serve as facilitator and consultant in the objective assessment of risks of various technologies or approaches throughout the development cycle.

The industrial counterpart to the military PEO is the senior executive (typically division or site manager) responsible for the development of the contracted system. The SEI promotes the acceptance and use of methodologies that bring a higher degree of management control to the contractor's development processes while reducing the degree of technical risk at crucial points in the development cycle. The guiding principle for the SEI is the belief that acquisition and development should function more as a partnership than as a traditional adversarial business venture because the resulting system frequently involves life-critical functions supported by technology that is important to the nation as a whole; it is not a simple profit-seeking activity for the benefit of a single business enterprise. Therefore, the SEI promotes the best practices for effective management by both parties to the contract.

Practitioners are responsible for both pre-deployment and post-deployment total lifecycle software support. Government software engineering centers provide technical support to the PEO throughout the acquisition phase and the end item manager for the remainder of the system life cycle. These centers assist the PEO in ensuring that the software being developed for the system can be supported by internal resources or contractor support. Once the system is operational, the center is then responsible for software development support through enhancements and refinements that generally result in software version changes on a cyclic basis. LCSS practitioners work with the PM during development to ensure software product quality. These government practitioners must acquire a level of technical expertise that will give them sufficient knowledge to monitor contractor processes, methods, and tools that are employed to develop a given software system.

The SEI brings its effort to bear on identifying, evaluating, and disseminating methods, tools, and techniques that suggest a significant improvement over traditional approaches to system development. The intent is to bring to the acquisition team a greater ability to articulate software system requirements and to have developers equipped with the best knowledge and technologies currently available. This dual support for the acquiring government agency and the industrial contractor represents the best way to affect the overall set of lifecycle concerns.

Support for software development activities includes providing the practitioner with methods that ensure consistently high quality results in terms of system performance. Such methods address issues ranging from requirements analysis through design, coding, and test and integration. Further support to the practitioner comes in the form of tools that help automate certain aspects of particular methodologies. The goal is to provide the practitioner with an integrated set of methods and tools that will enable consistent results for the individual, the project team, and the suite of projects within the parent organization.

Post-deployment support generally corrects latent defects and performs enhancements to add greater functionality for incorporating new requirements to the existing system. The support centers are augmented with support contractors that will provide software development for each system that is in post-deployment. This contractor service augments the government practitioners who are responsible for the development of the version and block changes to each system. Practitioners that work within the PDSS community must benefit from the application of disciplined software engineering and the creative use of existing technology. Reengineering will be a new process that will help to improve productivity within PDSS.

Educators (and trainers) are responsible for meeting the nation's need for well-qualified software engineering professionals. In addition, education and training are essential components of technology transition.

By charter, software engineering education is part of the SEI mission. To better prepare new and existing software engineers to perform high-quality software development, the SEI must accelerate the development of software engineering programs in academic institutions. However, the education need is not limited to the academic sector. To improve the capability of current software practitioners, the SEI must enhance the quality and availability of continuing education and training programs in government and industry. Our efforts will be successful to the extent that the education infrastructure prepares individuals to participate in the software engineering activities of our customers in industry and government.

2 Strategic Overview

Software has become critically important to both our national defense and economic survival. It pervades our entire society, providing more and more of the functionality previously provided by hardware and expanding the capacity of hardware for multiple applications. As a result, the strategic importance of the SEI mission to provide leadership in advancing the state of the practice of software engineering cannot be understated.

Our approach to improving software engineering practice is set in the strategic framework of maturing the software engineering profession (i.e., maturing the skills of the practitioners who develop and maintain software and of the managers who organize and lead these activities) by maturing the organizational and managerial processes and the technology to develop and maintain software. The strategic framework is unified by our core competency in the area of software technology transition and supported by our core competencies in the areas of software process definition, modeling, and measurement, and methods and tools for disciplined engineering of software systems.

This chapter describes the strategic factors that determine SEI plans and sets the context for their implementation in support of the SEI mission. Section 2.1, Situation Analysis, provides an analysis of the current political and economic situation, and describes the major trends that are projected to significantly impact the field of software engineering and the SEI over the next five years. Section 2.2, Strategy for Improving Software Engineering Practice, describes the strategic framework which unifies the SEI's activities in support of its mission over the next five years, its selected core competencies, and the rationale for their selection. Section 2.3, Planning Considerations, specifies constraints and briefly describes the evaluation criteria to provide a framework for selecting among competing priorities. Section 2.4, Conclusion, summarizes the strategic context that forms the basis for the SEI technical program described in Chapter 3.

This plan continues to be based on the assumption of a continuing DoD "no-growth" strategy for the SEI during the five-year planning period. It reflects a commitment to effective cost control, increased leverage of resources, and focused efforts in those areas that will provide the highest payoff to SEI customers. At the same time, the plan reflects the support of the SEI for evolving national priorities resulting from the changing world political and economic environment.

2.1 Situation Analysis

The dramatic geopolitical events of the early 1990s and the changes in national focus being implemented by the Clinton administration will result in several major trends, discussed below, in the current political and economic environment. These trends will impact the SEI technical plans significantly during the five-year planning period.

National priorities have been influenced significantly by the dissolution of the Soviet Union and the increased boldness and military capability of third-world countries. The changing and undefined regionalized military threat, along with the renewed focus on national competitiveness at the global level, are impacting the emerging policies of the Clinton administration. These policies are reflected in the reduction of DoD budgets, the downsizing and changing role of the military, and the increasing concern for the domestic infrastructure and our global competitiveness. To implement these policies, the administration has developed the National Technology Policy (NTP), changed the role of the ARPA, and initiated the Defense Conversion Program, the Technology Reinvestment Project, and the National Information Infrastructure (NII).

In the face of the new world order and the shifting focus to domestic economic issues, the goals of the Clinton Administration include increasing investment in the national industrial base, revitalizing the national infrastructure, improving education in math and science, protecting the environment, shifting defense-based research and development programs to the industrial sector, and achieving global competitiveness. Clearly, supporting these goals will have significant influence on the long-term direction and technical focus of the SEI.

2.1.1 Shrinking DoD Budgets and Downsizing of the Military

Budget and force reductions will result in a decrease in DoD organic capabilities, a growing need for increased flexibility, concern for system evolution, and a smaller DoD contractor base. All of this will lead to pressure to maintain existing systems and components for longer periods. Fewer new systems will be built, existing systems will have to be evolved to meet new threats, and simulation will become an even more cost effective method for military training and system evaluation. Since the DoD will be a customer with less spending power than in the past, it will have reduced influence on the strategic directions of industry. This reduction in influence will create additional reasons for the DoD to make maximum use of commercial products and will create more dependence on dual use technology, reuse, and reengineering for extensive and responsive system modifications.

2.1.2 The Changing Role of the Military

The decline of the Soviet Union as a military superpower has been accompanied by the growing military capability of many third-world countries. More of these nations may be able to develop or acquire nuclear capability, medium-range missile delivery capability, and chemical or biological weapons. The political instability in some of these countries gives cause for alarm,

as does the fact that extra-national groups (e.g., those involved in illegal drug trafficking) are increasingly well funded and equipped, and are willing to engage in military and political activities. The undefined nature of this regional threat will require the rapid development and deployment of systems to support the changing mission of the military to one of rapid response to both military threats and humanitarian needs, and will require a refocusing on the need for improved logistic support and command and control. At the same time, recent domestic and international natural disasters have reemphasized the potential role for the military in providing humanitarian relief and assistance. This role will require improved humanitarian logistic support and domestic emergency management command and control.

2.1.3 The Increasing Concern for the Domestic Infrastructure

The NTP has been developed to rejuvenate the nation's economic infrastructure and to revitalize our national industrial base. This policy will increase the focus on research and development (R&D) and technology transition in the commercial sector, development of information technology as an engine of economic growth, and improvements in manufacturing technology, health care, transportation, communications, and education. ARPA and the Department of Commerce, through the National Institute of Standards and Technology, will play an increasingly important role in supporting this focus through the development and deployment of emerging dual use technologies.

The increasing importance of software engineering in support of the National Technology Policy's focus on technology transition for improved national competitiveness will most likely shift the national R&D focus toward improving manufacturing technologies, upgrading the national transportation system, implementing the NII and enhancing commercial communications, and addressing software reliability in critical medical applications for patient monitoring, treatment, and medical imaging.

2.1.4 The Increase in Global Competitiveness

The increased focus on national competitiveness at the global level will strengthen the importance of international standards and highlight the increasing global technological sophistication. This will require increased participation by the SEI in the international technical community to assist in the development of the standards required for doing business in this environment, to interpret the impact of these standards on our own economic base, and to assist in the formulation of a responsive national policy for effective competition in the global marketplace.

2.1.5 Effect of Major Trends on Software

During the next several years, the federal budgeting process will most likely shift to reflect the changing national priorities evidenced in the four major trends just described. As a result, we expect to see no more than the same level of funding support from the DoD. At the same time, we can expect increased need and funding support from other federal agencies as they struggle to overcome the same software-related problems that have been addressed effectively within the DoD over the past decade.

Defense planners will have to create a smaller and better-trained force supported by high-performance equipment that can be adapted to changing threats. This equipment and the systems that support better training are increasingly dependent upon software for their functionality. Concurrently, computing power, resulting from advanced semiconductor technology, continues to double about every four years. These trends are creating demands for affordable, reliable, and flexible software that are becoming more difficult to satisfy.

In light of the changing federal budget and the increased importance of dual use technologies that satisfy both defense and commercial needs, it is possible that the SEI could be supported by multiple federal sponsors. In addition, as the DoD is downsizing and emphasis on improving the U.S. economy and infrastructure moves the nation toward a more commercially oriented R&D base with focus on technology transition to the commercial sector, increasing amounts of research relevant to the DoD will be conducted by other federal agencies or industry. Hence, the SEI must pay increased attention to providing support to other federal agencies and industry to participate in this broadened range of dual use technology developments relevant to DoD software engineering needs.

Increased national competitiveness will require shortened system development times to meet unforeseen requirements. This suggests that future systems will be created and configured on demand from proven concepts, architectures, and components. This situation will place greater emphasis on software architecture, reuse, and reengineering in the shorter term and design for reengineering and automatic program generation in the longer term. It will also require more effective software engineering practices that can be applied much earlier in the system life cycle than is now the case.

The reduction in operational funding for the military will also emphasize the importance and cost effectiveness of simulation for training. The increasing use of real-time simulation, both for defining and refining system requirements as well as for training, suggests an increasing need for software that can meet time constraints within a network environment. It also suggests the need for vastly improved human interface technologies to provide the realism needed for effective evaluation and training.

The need to respond quickly anywhere in a worldwide theater of operations suggests increased portability of command-control and intelligence facilities. It also suggests greater use of concepts such as teleconferencing and telepresence so that people with critically needed skills, but who are located remotely, can be brought to bear in solving local battlefield or humanitarian relief problems.

Use of products and standards emanating from the commercial world offers an attractive way for the DoD to acquire and evolve systems of high quality at minimum cost and risk. DoD system acquisition procedures may change to allow more frequent use of commercial off-the-shelf (COTS) products. In using COTS, industry standards will become even more important to the DoD. Hence, the SEI must focus on the capability to design systems and their architectures using these standards.

Budgetary constraints and rapid changes in military strategy are also creating the need for more flexible manufacturing capability. DoD acquisition may fund prototype development with full-scale production deferred until its need is clearly demonstrated. The dual use nature of this emphasis on "agile manufacturing" for both defense and commercial needs implies that the SEI devote more attention to processes and tools for supporting manufacturing technology transition efforts.

In summary, the SEI must respond to the changing political and economic environment and the need to revitalize the national infrastructure, and particularly the nation's industrial base. This response should increase our focus on dual use software engineering technology. The SEI should support the development of enhanced software architectures, improved techniques for reuse and reengineering, real-time simulation, expanded and more flexible communications capabilities, system integration of commercial software, and software engineering processes and tools for advanced manufacturing technologies. To develop the underlying foundation for our technical focus areas, work should continue in education and training and improving the state of the practice of software engineering through improved processes and software risk management.

2.2 Strategy for Improving Software Engineering Practice

The current political and economic situation, as described in Section 2.1, clearly establishes the importance of software to the defense and economic well being of the nation. Because software pervades nearly every aspect of society, it is vital to address effectively the issue of continuous improvement of the practice of software engineering as an essential ingredient of our national strategy. With this in mind, it is important to articulate clearly the strategic framework which supports the SEI mission to improve the state of the practice of software engineering.

Figure 2-1 summarizes the SEI strategic framework to improve software engineering practice in support of our national strategy. It is through this strategic framework that the mission of the SEI will be executed effectively.

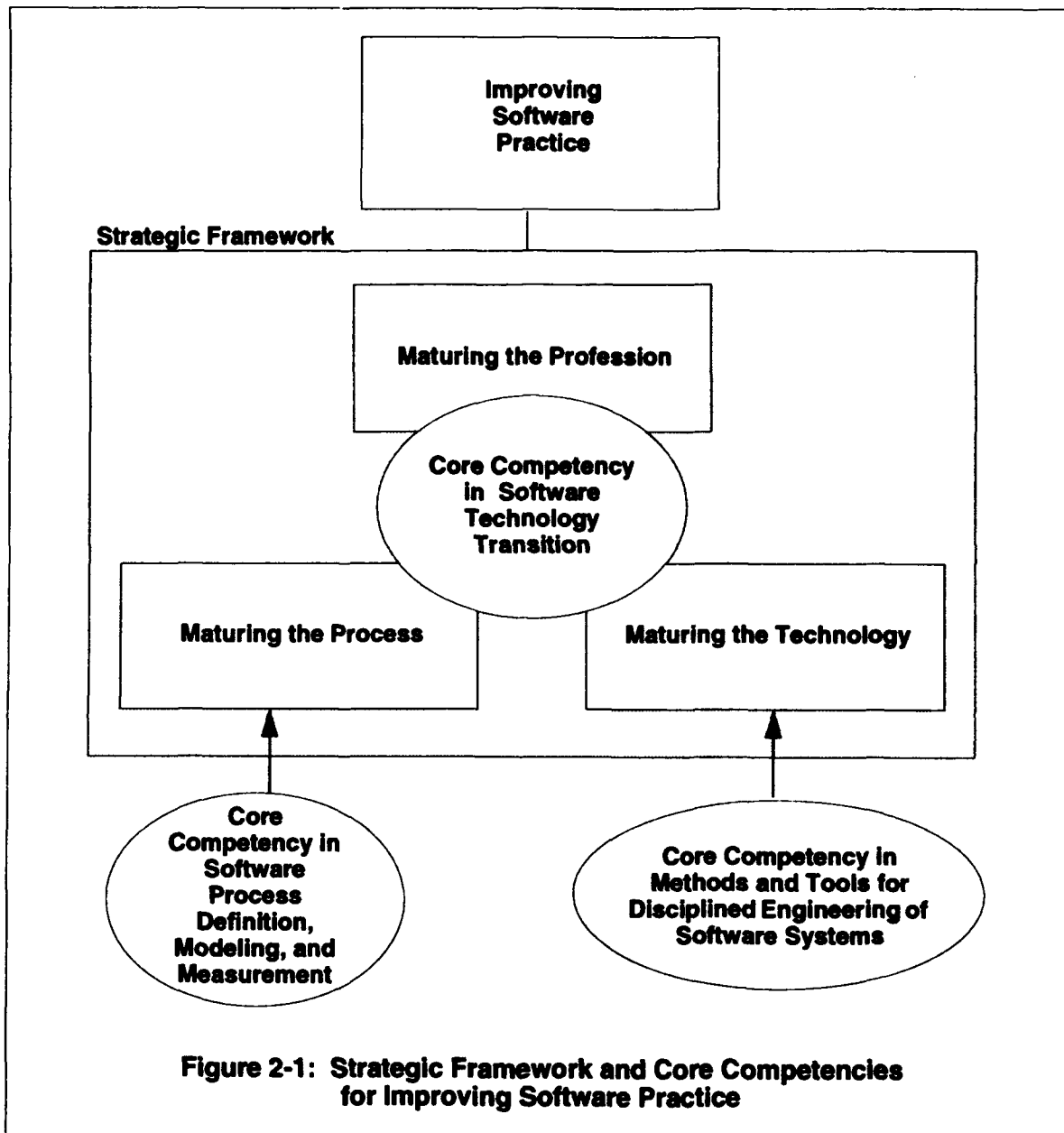
The SEI strategy for improving the state of the practice of software engineering is to mature the software engineering profession (Maturing the Profession). This strategy is based on maturing the skills of the software engineering practitioners who develop and maintain software and the managers who organize and lead these activities. Our approach to improving the skills of these software engineering professionals is to mature the organizational and managerial processes through which software is developed and maintained (Maturing the Process) and the technology used to develop and maintain software (Maturing the Technology). These activities, unified by our core competency in transition, form the strategic framework for the execution of the SEI mission.

The SEI has chosen to develop and maintain three core competencies:

- Core competency in software process definition, modeling, and measurement
- Core competency in methods and tools for disciplined engineering of software systems
- Core competency in software technology transition

Of these, our core competency in software technology transition has been included in the strategic framework because of its centrality to the SEI mission. Our core competencies in process definition, modeling, and measurement; and methods and tools for disciplined engineering of software systems provide support for this strategic framework from within the technical program described in Chapter 3.

Each of these core competencies is described in detail in the following subsections.



2.2.1 Software Process Definition, Modeling, and Measurement

By software process definition, modeling, and measurement, we mean those organizational and management activities performed by people operating within an environment through which software is defined, developed, and maintained. We chose to develop software process definition, modeling, and measurement as core competencies because no uniform, defined, and measurable process for effectively organizing and managing the development of software for complex systems existed within the software development community of the federal gov-

ernment. The SEI approach is to define this process, develop models of it, and determine how to measure its effectiveness. Our objective is to improve the process so that high-quality software can be developed on time and within budget. By high-quality software, we mean software that meets the user's needs and that can be affordably maintained. The SEI is widely acknowledged as the leader in defining, measuring, and modeling software process. More information on process definition, modeling, and measurement can be found in Section 3.2.

2.2.2 Methods and Tools for Disciplined Engineering of Software Systems

By methods and tools for disciplined engineering of software systems, we mean those techniques and software systems that operate upon and within the process described in Section 2.2.1. Our emphasis is on methods and tools at a meta level both with respect to the methods and tools themselves and to their application. For example, we are more concerned with applying methods and tools that maintain engineering information (such as domain models, architectures, and their quality attributes) than in developing and refining particular algorithms. We are more concerned with identifying ways to evaluate, specify, integrate, and adopt methods and tools (such as computer-aided software engineering (CASE) tools) than in developing or promoting particular implementations. More information on methods and tools can be found in Sections 3.4 and 3.5.

2.2.3 Software Technology Transition

By software technology transition, we mean movement of the best software engineering processes, methods, and tools from research and development into broad use in the software engineering community. The SEI is a value-adding transition agent between researchers whose results can improve software practice and practitioners who can apply this result to solve important and pervasive software problems. The SEI adds value by identifying relevant research results and making them understandable and applicable by practitioners, and by identifying root causes of problems faced by practitioners and making them understandable and applicable to researchers. Thus, while the computing research community aims to advance the state of the art, the SEI aims to incorporate state-of-the-art advances into the state of the practice. Through our interactions with practitioners and researchers, the SEI seeks to identify "best practices" and to promote widely their introduction into the practice of software engineering. Successful technology transition results in overall improvement in the state of software engineering practice.

To build its core competency in transition, the SEI adapts transition models from other disciplines and applies them to software. These models help us approach technology transition in a systematic and effective way. We identify transition methods and transition vehicles that facilitate adopting and institutionalizing improved processes, methods, and tools. We develop transition products and services that help people help themselves improve their practices.

Because the size of the SEI technical staff is limited, we seek leverage for our transition efforts. We gain leverage by influencing software engineering education and providing educational materials that aid the teaching of good software engineering practices, by providing services—primarily advice and guidance to government organizations—that aid continuous improvement efforts, and by working with transition partners who can take our products and services to the community at large.

2.3 Planning Considerations

In developing plans within the context of the changing political and economic environment, the SEI must consider several factors. The most significant of these planning considerations are:

- The mission requires that the SEI facilitate the transition of appropriate software technology into practice for the mutual benefit of the DoD and other federal agencies in support of national priorities and objectives.
- The SEI strength is in the area of software technology—broadly defined to include traditional “computer science” and the evolution and transition of engineering practice. The SEI must maintain its focus on technology to provide the stability that is vital to an R&D program, and must emphasize efforts that result in products rather than personal services.
- The small size of the SEI requires highly leveraged resources and a very effective means of technology transition.
- Technology transition requires knowledge of, and involvement with, technologies, user communities, and the transition process. While most SEI activities focus on software technology, they have a planned “side effect” of increasing SEI knowledge about user needs in specific domains.
- The SEI must balance technical depth with a broad understanding of software practice in its personnel.
- The SEI must maintain its position as an objective third party to function effectively as a center of excellence in software engineering technology.
- The SEI must act under the assumption of a DoD “no growth” strategy and plan for the expansion of resources to support the objectives of the National Technology Policy through sponsorship by those federal agencies charged with its implementation.
- The SEI must not violate contractual constraints prohibiting competition. Our approach to transition, which seeks to use the existing U.S. infrastructure as partners, helps us to avoid competition.

In addition to these planning considerations, the SEI must carefully consider additional evaluation criteria in order to maximize the benefits to our customers within the context of the major trends affecting the software engineering profession. These criteria currently include:

- **Importance.** Does the result address an important software engineering problem?
- **Impact.** Does the result have potential for a significant impact?
- **Alignment.** Does the result align with the SEI mission?
- **Leadership.** Can the SEI exercise national leadership in this area?
- **Leverage.** Will the result give the SEI the leverage that is required for its fewer than 200 members of the technical staff to affect the work of practitioners in the software engineering community?
- **Pervasiveness.** How widespread is the need? Will the result be applied pervasively?
- **Transition.** Can the result be effectively and efficiently transitioned into practice?
- **Legacy.** Does the result contribute to the knowledge base that supports software engineering?
- **Synergy.** Does the activity provide synergy with other SEI and non-SEI efforts?
- **Likelihood of Success.** Is the activity likely to succeed?

2.4 Conclusion

The importance of software to our national security, both in terms of defense and economic well being, has never been more clear. As articulated in this chapter, the importance of software emphasizes the importance of the SEI mission to advance the state of the practice of software engineering. The SEI is firmly committed to this mission, and has established and implemented the strategic framework described in this chapter to achieve its mission. The foundation for this strategic framework is found in our technical program, described in Chapter 3. The goal of the SEI technical program is to improve software engineering practice by maturing the process, the technology, and the profession.

In responding to the challenges discussed in this chapter, the SEI seeks to take advantage of the organizational, historical, and situational differences that distinguish it. These differences include:

- Accomplishments to date, particularly in the areas of software process modeling, real-time systems, and software engineering education.

- The SEI status as a federally funded research and development center (FFRDC) chartered to act as an objective broker performing software engineering technology development and transition.
- The SEI association with CMU and its relationship to its world-class faculty in such areas as computer science, electrical and computer engineering, and economics and business.
- The SEI sponsorship by ARPA, which makes us a participant in the ARPA software research community.
- The SEI charter to provide research and technology support throughout the federal government, enhancing our ability to support the objectives of defense conversion and the NTP.

The SEI has established itself as the leader in the field of software engineering. We have developed and are implementing a strategic framework for improving the state of the practice of software engineering. In supporting this framework with a strong technical program and well-focused core competencies, the SEI has positioned itself to respond effectively to new requirements and technologies.

3 Technical Focus Areas

3.1 Overview

The goal of the SEI technical program is to improve software engineering practice by:

- Maturing the skills of practitioners who develop and maintain software and of managers who organize and lead those activities (Maturing the Profession).
- Maturing the organizational and managerial processes through which practitioners develop and maintain software (Maturing the Process).
- Maturing the technology used by practitioners to develop and maintain software (Maturing the Technology).

These three strategic activities—maturing the profession, maturing the process, and maturing the technology, combined with the SEI core competency in software technology transition—comprise the strategic framework that guides SEI technical efforts (see Figure 2-1).

In this chapter we describe the SEI technical program and show how it will improve software engineering practice through the strategic framework by increasing the maturity of the profession, the process, and the technology.

3.1.1 Maturing the Profession

Extremely instrumental in achieving a sound, strategic framework for maturing the profession are the efforts of software practitioners and managers, their respective skill levels, and their professional work environments. This maturing process and its contributors are acknowledged in Figure 2-1. For example, many SEI products and services, described in Chapter 4, serve as tools in attaining this maturity via SEI products for software practitioners, managers, and educators.

We believe that it will be possible to define and measure the maturity of the profession. However, due to limited resources, we will only begin feasibility efforts in 1994 to define and measure this maturity.

3.1.2 Maturing the Process

The capability maturity model (CMM) was conceived as a model for judging the maturity of the organizational and managerial processes of an organization and for identifying the key practices that are required for maturing these processes. It is proving to be very effective in the commercial as well as defense software communities.

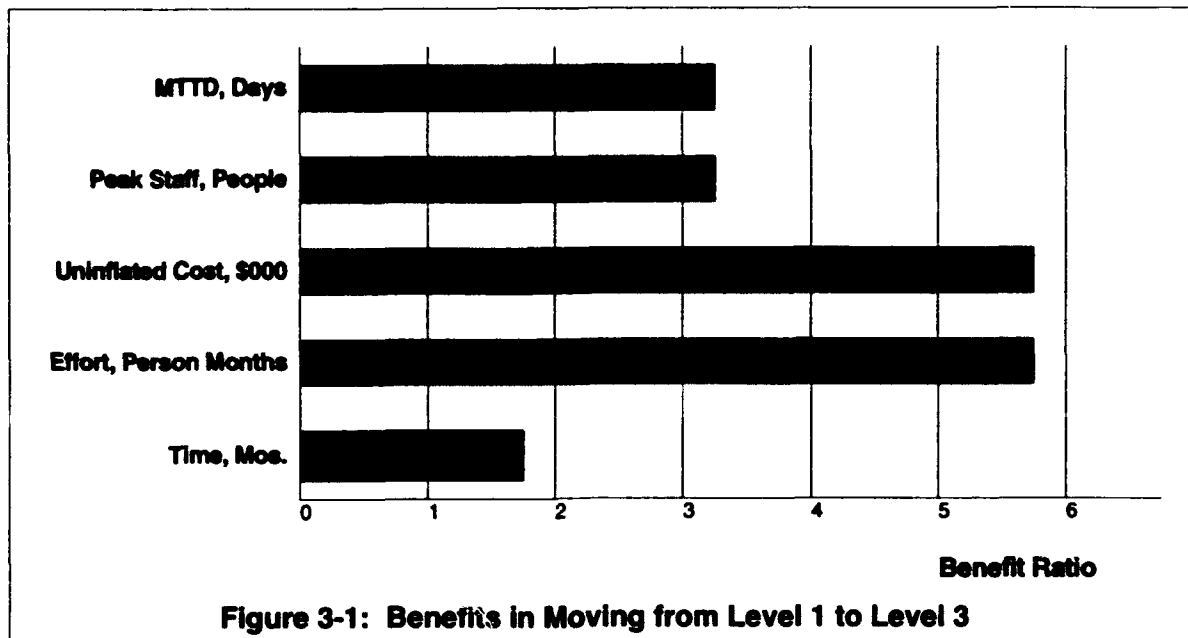
For example, Wohlwend and Rosenbaum [Wohlwend 93] of Schlumberger Laboratory reported experiences with CMM-derived improvements from 1989 to 1993. A Schlumberger group that works on complex embedded real-time systems reported that they were able to reduce the number of validation cycles from 34 to 15. This, in turn, reduced time to market. The group credits this improvement to the introduction of requirements management, a level 2 key process area (KPA).

Wohlwend and Rosenbaum also reported that one engineering group began concentrating on process improvements in mid-1990 and improved on-schedule completion from 51 percent in 1990, to 89 percent in 1991, and to 94 percent in 1992. The group credits this improvement to improvement in initial project planning, another level 2 KPA. For each level 2 KPA introduced, Wohlwend and Rosenbaum describe similarly impressive improvements.

Lipke and Butler [Lipke 92] of the Oklahoma Air Logistics Center (OC-ALC) described efforts that began in 1989 to introduce SEI methodologies into the work environment of the Aircraft Software Division (LAS) of OC-ALC. To guide their process improvement efforts, LAS established a Quality Management Steering Team and a Software Engineering Process Group (SEPG).

The LAS SEPG wrote an action plan for improvement that is periodically updated and revised. The purpose of the action plan is to set improvement goals; discuss how improvement efforts will be measured; and provide a template for planning, developing, and implementing improvements. As of November 1992, LAS had introduced 44 improvements and had gathered return on investment (ROI) data on 18 of these. For the 18, \$2.935 million was returned by an investment of \$462,100, resulting in an ROI of 6.31.

Most recently, Putnam [Putnam 93] reported results of moving up the SEI CMM scale from a high level 1 to a low level 3 from 1988 to 1992 for one system done at a central design agency within the DoD. Figure 3-1, taken from Putnam's report, shows ratios of improvement from 1.7 to 5.7 in measures such as mean time to defect (MTTD), peak staffing, cost, effort, and time. These are consistent with previously published improvements at Hughes [Humphrey 91], Raytheon [Dion 92], and Hewlett-Packard [Grady 89].



3.1.3 Maturing the Technology

In 1993 the SEI initiated activities to consider whether it would be possible and practical to define, measure, and mature technology as was done for maturing organizational and managerial process through the CMM. An SEI task force was commissioned to address this issue, and resources were identified to support this effort at a low level in CY 1993. Work to date suggests that a maturity model similar in principal and intent, along with key techniques similar to key practice areas, can be defined. It is too early to report on the specifics of this activity. Since this appears to be a fruitful area for further investigation, funds for this activity are being allocated in CY 1994.

3.1.4 Technical Focus Areas

We have organized our technical program into clusters of related activities called focus areas that are responsible for identifying and transitioning those technologies that we believe will mature the profession, the process, and the technology. The four focus areas are: process, risk, methods and tools, and real-time systems. Work conducted in the focus areas also maintains the SEI core competencies in software process definition, modeling, and measurement; methods and tools for disciplined engineering of software systems, and software technology transition.

The relationship of the focus areas and core competencies to the SEI strategic framework is shown in Figure 3-2.

Our reason for choosing these four focus areas derives from our belief that to be successful in improving our customers' software practices, software organizations must have a well-defined process supported by methods and tools used by appropriately educated and trained professionals. Definitions of these terms are:

- A **process** is a series of activities, transformations, or procedures that achieve a desired result.
- A **method** is a systematic approach to performing work during the execution of a process.
- A **tool** is an instrument that provides support for a method. (An automated tool is software that provides automated support.)

Each step in this hierarchy of process, methods, and tools can itself be a process supported by its own methods and tools, although this is not essential.

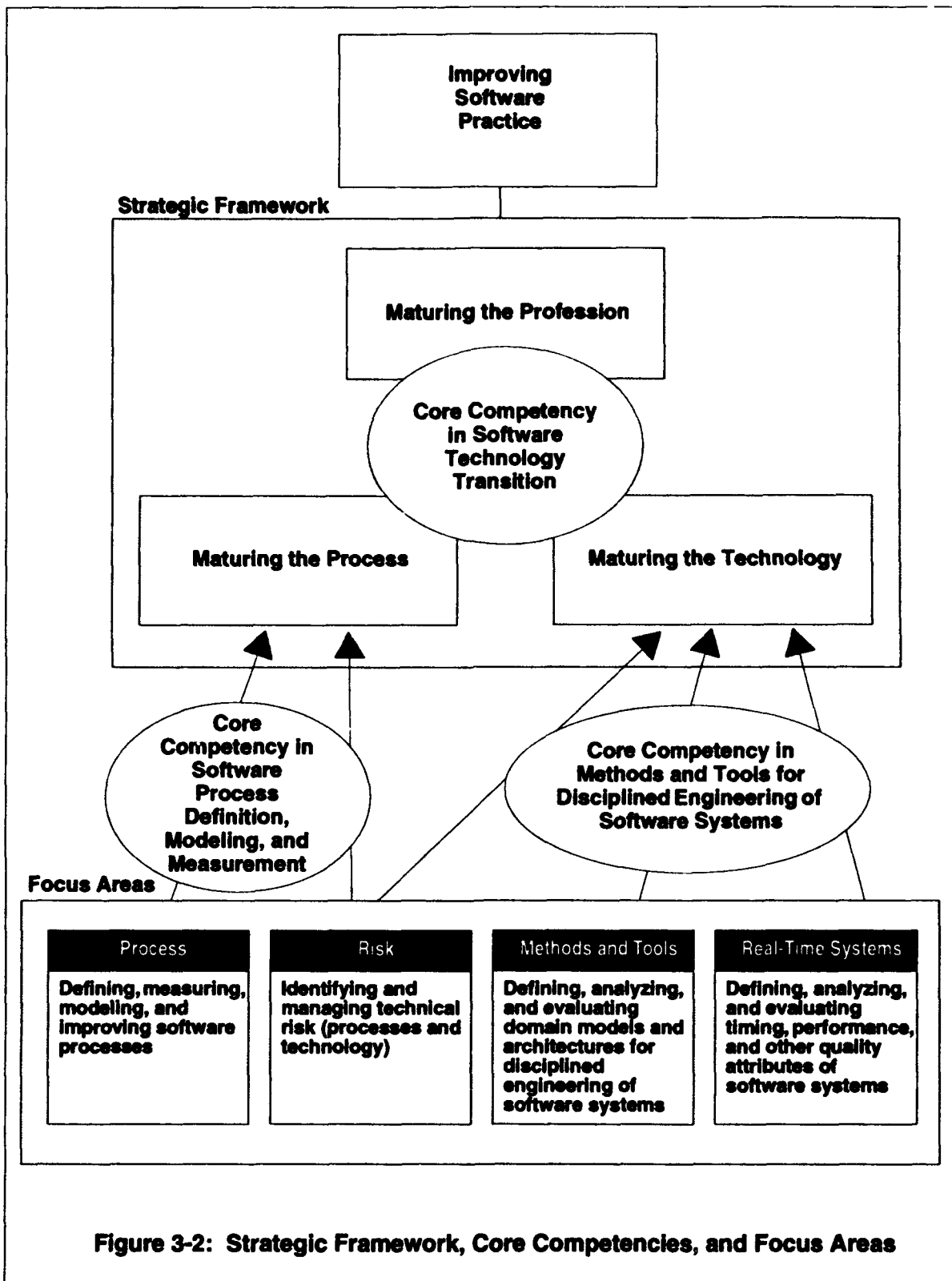
Our focus on process is presently concerned with the maturity of the organizational and managerial processes employed by software developing organizations. The SEI seeks to define, model, measure, and improve the maturity of these processes. The expectation is that doing so will improve the organizational performance in developing software. As discussed in Section 3.1.2, this expectation is being satisfied.

Our technical focus on risk provides a systematic and structured process, supported by methods and tools, for identifying and analyzing the technical uncertainties encountered in a specific software development. Having such a process will significantly enhance the probability of success of a program by allowing risk-resolving steps to be taken before problems occur.

Our next focus area is developing methods and tools that operate on or within the software for defining, analyzing, and evaluating domain models and architectures for the disciplined engineering of software systems.

Our focus in real-time systems is motivated by the need for systems that must satisfy critical real-time constraints. Our goal has been to provide methods and tools for creating such systems that can guarantee that timing constraints are met. Our goal is now broadened to address dual-use (defense and civilian) applications where quality attributes such as timeliness, reliability, safety, and interoperability are important.

Each SEI focus area is presented in depth below (in Section 3.2 to Section 3.5), including the problems being addressed, the approach to solving them, and potential products that can result from their solution.



3.1.5 Measures of Success

The measure of success collectively associated with these technical focus areas is the extent to which quantifiable returns on investment (ROI) can be related to increases in the maturity of process, technology, and profession. As mentioned in Section 3.1.2, a body of experience with the use of CMM-related process products and services is beginning to provide credible and quantified evidence that high ROIs can be correlated with increases in process maturity.

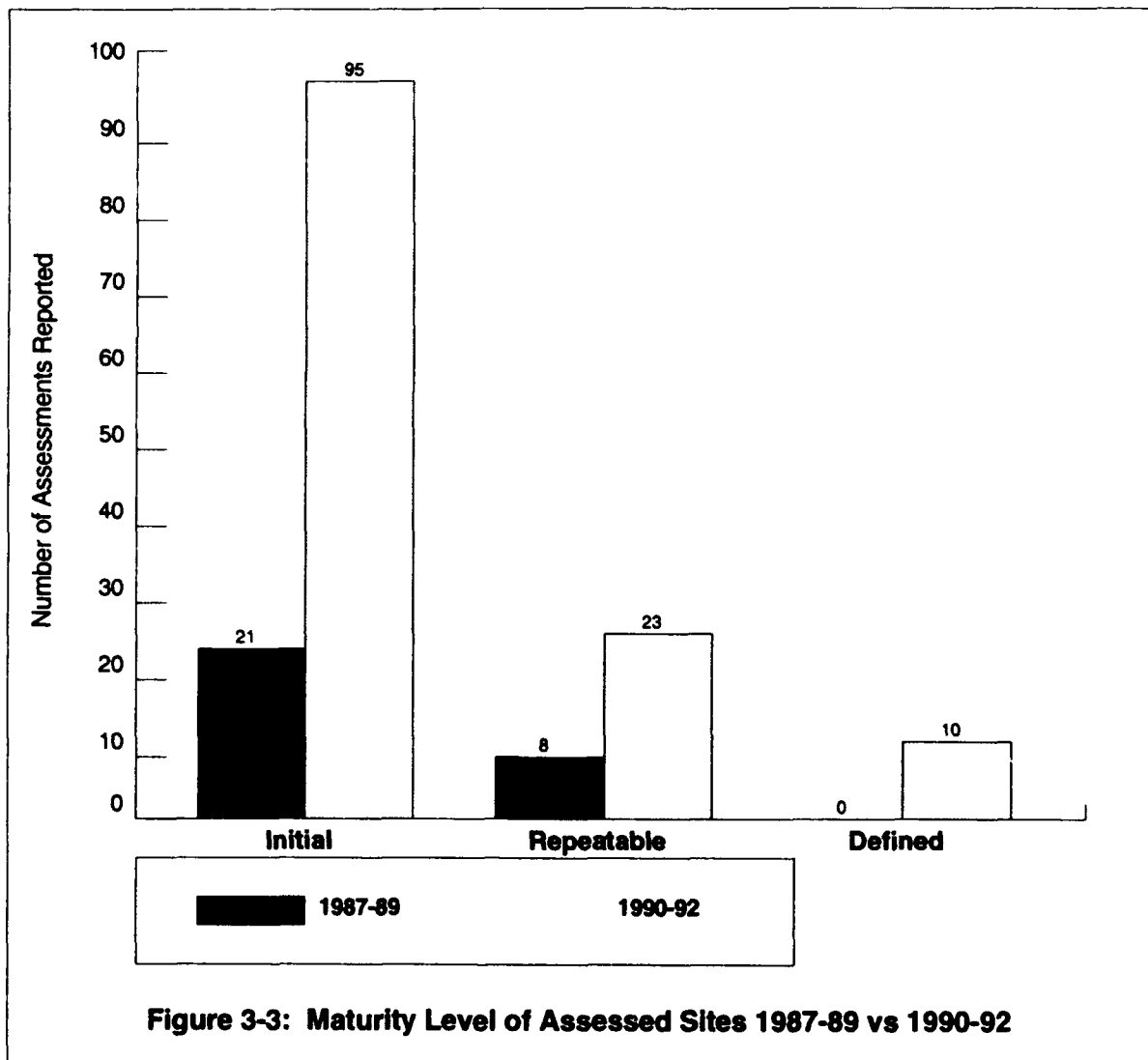
As yet, there are no defined levels of technical or professional maturity to support a similar ROI analysis for the profession and the technology, although we have plans in place to define these for the technology and are considering them for the profession. In the meantime, we will use technical objectives and plans (TO&P) report-card scores, community interest as expressed by targeted conference or course attendance (for example, the Risk Conference), and potential products transitioned into planned products as surrogate measures of success.

3.1.6 Effects of Unpredictable Funding

This plan is not based solely on core funding, but instead assumes a mix of TO&P, core, and cost-recovery funding. Core and cost-recovery funding, compared with TO&P funding, are relatively predictable. For TO&P funding, we seek customers willing to fund activities that most nearly fit the program we have planned. Sometimes we succeed; sometimes we do not. Other times we find TO&P customers whose needs closely enough align with our overall plan that we are able to address them. In this latter case, we alter our plan to provide as much improvement in practice as the combined funding allows. Thus, while we intend to execute the plan presented here, the plan that we will actually execute is to a large degree dependent upon the TO&P funding we can attract.

3.2 Process

Large numbers of software development organizations in the United States continue to have an ad hoc, crisis-driven process that often results in projects producing poor-quality systems that are chronically late and over budget. While organizations are improving their process capability, as assessment data show, most of the software organizations are at the lowest level of software process maturity. See Figure 3-3. This data was based on SEI-assisted, vendor, and self assessments.



These crisis-driven software environments are characterized by:

- unpredictable, inconsistent performance and quality
- an inability to successfully implement and sustain new technologies and methods
- strong dependence on a few highly competent individuals
- a focus on fire-fighting
- high levels of frustration and adversarial relationships across disciplines
- predominantly schedule driven environments

Too many software organizations have long relied on individual talent that is in short supply to produce acceptable results. Unfortunately, the only way these results can be repeated is to assign the same individuals to the next project. In such environments there is no institutionalized capability for meeting projected targets for cost, schedule, quality, and functionality. Executives in such organizations typically complain that they have little visibility into the software development process, and they are unable to make accurate projections of performance and costs. Without visibility and predictability, executives are unable to exercise management oversight or to make sound business decisions regarding projects.

To improve the state of the practice of software engineering in the U.S., software producing organizations must establish an organizational capability (rather than be dependent on individuals) for developing software based on sound management practices that support a disciplined, defined, and measured software engineering process. They must be able to execute this defined engineering process consistently across all projects in the organization, rather than have only a few successful projects, with others missing the objectives of quality, cost, function, and schedule.

3.2.1 Five-Year Plan

3.2.1.1 Goals

A primary objective for the SEI process focus area is to present to the software community an integrated approach to software process improvement. An organization should be able to look to the SEI for a suite of products and services to determine where and how it should start and continue along the path of continuous process improvement for software development.

A goal of the process focus area in last year's plan was that by 1998 all major software producing organizations that serve the DoD community would have instituted continuous process improvement programs. We now expect that by 1999 such programs will have been instituted not only by the defense industry but also by commercial software producing organizations.

Achieving these goals should make U.S. software developers among the most competitive in the world in terms of cost, quality, and time to market. This focus on process will generate requirements for process technologies that can be incorporated into software engineering environments.

Another goal in last year's plan was to see a strong commercial infrastructure developed within the U.S. by 1998 to provide inter-company communication about lessons learned in process improvement. We expect that this infrastructure will be more deeply grounded by 1999 where wide competitive choices will be available to satisfy the needs of the software development community. A sufficient cadre of trainers and consultants will be in place for educating customers about organizational change and process improvement and sustaining customer improvements. Efforts to promote a commercial process improvement industry with quality standards for practitioners through such organizations as the International Standards Organization (ISO), for example with ISO9000 and SPICE (Software Process Improvement Capability dEtermination), will continue, and we will work to ensure that the SEI and U.S. perspectives for software process improvement are addressed in such standards.

SEI process improvement programs are planned to dramatically improve the quality and reduce the costs and schedule slippage of software developed for the DoD and other government software producing and procuring agencies. We expect Software Process Improvement Network (SPIN) groups to increase to at least 10 national locations. We also expect each SEPG national meeting will continue to attract in excess of 500 participants.

Our goal last year was to lead a national commitment to improvement and a service business to support it, so by 1998, 80 percent of all defense contracting sites with more than 50 software engineers will have active process improvement programs, and 50 percent will have advanced one level more from their initial measured maturity level. We expect that by 1999 this goal will address other government software producing agencies as well.

Productivity, quality, and process measures will be routinely collected and used to improve the software process in those organizations. A sufficient number of contractors with a defined, measured, and managed software process will exist by 1999, so the DoD and other government agencies will not need to use contractors with weaker software process capability for software intensive systems costing \$10M or more. Further, by 1999 all software systems procured from contractors with high process maturity (Managed or Optimizing) should be delivered with fewer than one-tenth of a defect per thousand source lines of code. This is a significant improvement over the current goal of one defect per thousand source lines of code, which is what many industry software groups have set as their target in 1993.

The overriding goal of the SEI process projects and products in this plan is to improve the maturity of software development organizations as defined in the CMM.

3.2.1.2 Strategy

Achieving these goals and being successful will require the SEI to continue to maintain and evolve the CMM. The CMM, developed at the SEI, describes five stages through which software organizations must pass to achieve a sustainable state of continuous process improvement. This five-stage model provides software organizations with guidance in planning a long-term process improvement program, in addition to assistance in setting priorities for immediate improvement activities.

A strategy of the process focus area is to maintain the CMM as a community-owned model for which the SEI provides stewardship until such time as some standards organization (e.g., ISO) adopts the CMM as the preferred standard. The CMM is constantly subjected to national and international review, has recently been revised as CMM Version 1.1, and is being accepted as a de facto standard for implementing process improvement activities. It is now common practice in technical literature to simply say "SEI CMM level 1" without explanation.

At the request of ARPA we are participating in DoD standards activities to bring a software perspective to such efforts as MIL-STD-SDD. A wide range of American software developers have urged the SEI to integrate CMM concepts into ISO standards. We continue to pursue these requirements. For example, the SEI is working with the ISO SPICE Project, which is working on international standards for software process improvement and capability determination.

To use the CMM for guiding process improvement activities such as those we are conducting at various agencies, such as the Army Materiel Command (AMC) and the Air Force Materiel Command (AFMC), we will continue to provide software organizations with methods to reliably assess the maturity of their own development and maintenance processes. Concurrently, we must provide acquisition organizations such as AMC, AFMC, and the Naval Air Systems Command (NAVAIR) with the ability to reliably evaluate the capability of their contractors.

Because of limited resources, the SEI must transfer the capability to train and execute these diagnostic methods to third parties. At this writing, the software process assessment technique has been licensed to 10 companies in the commercial sector and 2 DoD agencies. We plan to expand this list to meet the needs of the software producing community with market competitive pricing for such services. In 1993, we began to transfer the ability to train capability evaluation teams to training sources inside the DoD such as the Defense Systems Management College (DSMC).

To affect growth in the national software capability, these diagnostic methods must be coupled with the ability to plan and implement process improvement programs tailored to the maturity level and specific problems of software organizations throughout the DoD and other government agencies. Since these evaluation and assessment methods are rapidly growing in use

and in importance in the software community, it is essential that certification be established during this strategic period. It is expected that certification will be fully established by 1995 for both the software capability evaluation (SCE) evaluators and the software process assessment (SPA) assessors.

After an assessment of process capability, an organization has great need for a defined process upon which to structure process improvement efforts. Particularly for level 1 organizations, there is an uncertainty of the relative priorities of the activities necessary to start a process improvement effort. There are several products related to process definition that are essential to maintaining process improvement momentum:

- A defined process for process definition
- Examples of good software processes
- A library of proven process fragments
- Procedures for establishing requirements for a process
- A language for process definition
- Procedures for designing a process
- Procedures for implementing and installing a defined process
- Procedures for evaluating a defined process against a standard
- Procedures for measuring the effectiveness of a defined process in action

Collecting technology for these products, creating the products, and getting them into widespread use will be a major activity during the coming five-year period.

The SEI supports the development of process improvement programs throughout the defense and civilian sectors. To introduce these programs broadly in both sectors, we must continue to support the development of SEPGs and to educate them in the skills and tactics needed to successfully execute improvement programs. The SEI supports these SEPGs by providing them with specific methods and training designed to enhance software process improvements. In executing an improvement action plan, SEPGs need methods for defining and measuring software processes. We need to continue to develop, evolve, and pilot these methods and determine that they can be installed in the context of an improvement program. While we are conducting pilots, such as those at the Naval Air Warfare Center (NAWC) and AMC, we must develop courses based on our direct experiences to prepare others to incorporate these methods in their improvement programs.

Organizations are beginning to ask how they can estimate cost savings or quantify improvements before they introduce new software process improvement approaches. The SEI needs to provide such a means of estimation. This work will start in late 1993 and continue through 1998.

The process focus area has six advisory boards or steering committees with skilled and qualified representatives from the academic, industry, and government sectors. These groups will continue to provide advice throughout this strategic period. It is our intent to have minimal duplication of focus between these groups and to ensure that they are kept up to date on each other's advice to the SEI process area. The following groups presently exist:

The Software Process Advisory Board oversees process products and their supporting projects. It provides advice concerning current and future strategic directions of process products. Board meetings are held three times a year. Board members were carefully chosen for their expertise and experience; they consist of:

- two members from the DoD
- two members from the DoD contractor community
- two members from academia
- one member from CMU
- two former SEI Process directors

The CMM Advisory Board reviews proposed enhancements to the CMM and related products prior to their release; provides written recommendations to further the SEI mission and user satisfaction for the CMM and CMM-based products; provides advice on CMM product objectives and development plans; and facilitates acceptance of CMM products by the user community. The eleven member board consists of the CMM project leader, the Process director, and at least three members from both government and industry organizations.

The Process Definition Advisory Group includes approximately forty members from industry, government, and academia who are leading researchers and practitioners in software process. The group provides a forum to define and debate process definition-related issues; refines objectives and evolves requirements; and reviews products.

The Software Process Measurement (SPM) Steering Committee provides technical input to the process measurement activities of the SEI. The steering committee is composed of twenty leaders in software measurement and management from industry, academia, government, and the SEI. The steering committee identifies and reviews measurement needs, activities and goals, progress, products, and future directions. The steering committee also promotes public acceptance of published results and work products.

The Software Capability Evaluation (SCE) Advisory Board has recently been formed. The role of the board is to act as volunteer consultants on the SCE method, its application, and its related products. The board consists of at least eight government members and eight industry members.

The **SCE Review Group** began operating in early 1993 and provides a "selectively broad" review of the SCE method and its documents and represents the software community. The 32 member group includes the advisory board, and members from small businesses, academia, and other groups in the SEI working on related products.

The SPA product set is presently receiving guidance from the SPA Vendors Association, a body represented by 10 industry groups. Additionally, we will draw the SCE and SPA efforts closer together by having a common advisory board between them.

Materials relating to SPA, including questionnaires, are reviewed by the SPA Vendors Association and the CMM Advisory Board.

3.2.1.3 Potential Products

The projects supporting work in the process focus area are closely interrelated, with each investigating a particular aspect of the larger problem. Within each of these subareas, a number of products will be produced over the next several years. Each of the products will be developed to discuss or explain an aspect of the methodology or to provide explicit instructions in the use of instruments contained within the methodology. Rather than listing those products individually here, the particular projects are discussed to clarify the issues. Individual work products are listed against the estimated time frame for delivery in Figure 3-4.

Capability Maturity Model (CMM) for Software (1Q and 3Q 1994, 1996/7). This model describes five stages in the maturation of a software organization's ability to manage its development process. Practices are not mandatory to achieve a level. Version 1.1 of the CMM was released in 1993, and will remain under maintenance for several years. The next upgraded and field tested version is targeted for the 1996/97 time frame.

There is little experience at levels 4 or 5. Therefore, additional key process areas may be added to these levels based on the effective practices required to support an organization at higher levels. We will begin work on level 4 and 5 enhancements in 1994. Examples of key process areas that could be added include risk management, reuse, and reengineering.

During 1993 we began studying how the CMM could be tailored for small organizations. The CMM tailored for small organizations will be delivered in succeeding years.

We also will begin development of human resources maturity aspects as an adjunct to the CMM to enable organizations to systematically increase their level of software talent. This addition involves improvements in recruiting, selection, performance management, training, compensation, career development, team and culture development, and organization design. After assessing the human resource capability for an organization, actions can then be taken

to improve these capabilities in line with the desired business needs for an organization. During 1993, a steering committee will be convened and an initial workshop will be held in late 1993 or early 1994. It is expected that the first version could be available in 1994 for pilot testing.

Software Process Assessment (SPA) Method (1Q and 3Q 1994). This diagnostic method offers software organizations an opportunity to determine where they are positioned relative to the maturity levels of the CMM. It is also an integral component of an improvement program, both in establishing a consensus on the problems facing an organization and serving as a catalyst in initiating action planning. The upgrading of the assessment method to use CMM v1.1 will be completed in late 1993 for maturity levels 1-3. Higher maturity levels may require different techniques. These would be made available beginning in 1994 and completed in 1995 after pilot explorations are completed. The upgraded assessment method and training will be available on a limited basis to selected software organizations and associates in 1993 and will be broadly available in 1994. This upgrade involves additional training and products to make the assessment method more repeatable and consistent across assessment teams.

The SPA method has been licensed to commercial organizations (e.g., the SPA Vendor Association) to increase the availability of assessment services. In the future, the SEI will expand the SPA vendor association to include additional associates and to address other process improvement needs. Additionally we are seeing a need for an assessment method for small organizations; this work will begin in 1994 and may be extended once we have determined the real need that exists across the software community. In future years, we will be expanding our work to include advances in process improvement action planning and implementation.

Based on current input, we may have assessment methods that can be executed in varying amounts of time, e.g., one day through several days, depending on the needs of an organization. The appropriate assessment method will be chosen based on organizational needs and constraints. For example, organizations that have performed full SPAs would like short snapshots during the course of their action planning and between full assessments to ensure that they are on target with their process improvement goals. Still other organizations are looking for a one-day scan to understand where they are with respect to the CMM. In all these cases, the full eight-day assessment is viewed as too costly. An abbreviated method set is needed to meet these user requirements. Each assessment vehicle will use the CMM framework, data collection, and analysis techniques and related products such as the questionnaires. Optionally, other tailoring techniques may need to be introduced during 1994.

Software Capability Evaluation (SCE) Method (1Q 1994, 1996/7). This diagnostic method allows procuring organizations to use the CMM to evaluate the risks of potential contractors and for contract monitoring for both the government and civilian sectors. Use of this diagnostic method in evaluating subcontractors and in monitoring requirements by both government contractors and prime contractors is expected to significantly increase during this period. Upgrad-

ing of the SCE method to use CMM v1.1 will be completed in 1993 and will be broadly distributed during 1994. This upgrade involves additional training and products to make the method more repeatable. Implementation techniques, practices, and user support materials are being collected into several guides that support the installation and long-term maintenance of this method. The ability to train this method will be transferred to selected DoD instructional facilities, other government agencies, and commercial organizations.

The SCE method and related documents will be improved incrementally and maintained until the next CMM is released in 1996/7. Piloting improvements with other government agencies (such as the National Oceanographic and Atmospheric Administration (NOAA)) and selected industry contractors using a structured transition approach as the framework for evolving the SCE method will result in expanded products and services. New industry partners will be sought to continue these efforts and refine the method for community institutionalization.

Software Process Definition (SPD) (2Q 1994, 1995). A prerequisite for orderly, sustained organizational process improvement is the definition of organizational processes. These processes are subsequently instrumented and deployed. Then at appropriate intervals, an analysis of qualitative and quantitative measures taken from these processes is used to formulate action plans to improve the process. This cycle is a fundamental process management concept and is dependent on achieving process fidelity or adherence in practice to the defined process.

Data from SEI software process assessments suggest that process fidelity (i.e., disciplined performance of specific actions) is low, and that many software processes are applicable only within the context in which they are being used. Current approaches to process definition lack the rigor and guidance that are required to produce a fit, usable process that will support the process management paradigm. Current process definition technology shares these deficiencies.

SEI software process definition research is guided by the needs of SEPGs, managers, and practitioners, from low- through high-maturity organizations. A collection of methods, guidelines, criteria, and models is being assembled to help organizations implement the documented, disciplined processes. These techniques cover the specification, design, tailoring, and implementation of improved software processes. They are based on an "engineering" approach that incorporates process management principles, process design guidelines and criteria, and process specifications derived from the CMM. They are supported by examples and instances of process architectures and elements.

As these techniques mature, they will be transitioned into general use through the evolving process technology transition infrastructure. Techniques for organizations with Initial through Defined maturity levels will undergo refinement and pilot testing during 1993 and early 1994. Training in these techniques and supporting materials is being offered to selected SEI strategic partners during 1993. A public offering is planned for 1994.

At the same time, advanced process definition techniques suitable for high maturity organizations are being investigated. Results will support the implementation of advanced process engineering capabilities. This work is addressing several state-of-the-art practices including the development of formal software process models, automated process analysis, and process enactment. Formal software process models based on graphical or textual languages with formal syntax and semantics will support automated analysis. Formalisms that support behavioral simulation (e.g., state machines or petri nets) can be used to analyze and evaluate the impact of a process change prior to implementation.

The planned and potential work described is facilitated through collaboration with a balanced mix of DoD, government, and industry partners.

Software Measurement (1Q and 4Q 1994). A collection of methods is being assembled to assist organizations in integrating measurement methods into their software process improvement programs. These techniques, tailored to the organizations' level of maturity, will be installed by project managers and SEPGs. During 1993, we began developing and pilot testing a workshop drawn from lessons learned from our measurement installation efforts. This workshop will be widely available beginning in 1994. Updates and improvements will be made to materials during the strategic period. At the same time, work will begin to probe new areas of measurement as requested by the community. These efforts include cost estimating techniques and other requirements at levels 4 and 5 in the CMM. Measurement definition checklists and frameworks will be developed for product measures. The current focus is on providing:

- Measurement definitions and methods for describing and reporting software measures that can serve as a basis for standardization.
- Guidelines, based on experiences, for establishing and sustaining successful measurement programs.
- Guidelines for integrating measurement with the CMM.
- Improvements in cost and schedule estimating processes for planning and managing software systems.

Empirical Methods and Studies of National Process Issues (4Q 1994, 1996/7). Tools to supplement SEI SPAs and support management tracking of process improvement are being developed collaboratively during 1993-94. For example, the Instant Profile, a rapid method to measure status between organizational assessments, is under development in collaboration with Pacific Bell and Hewlett-Packard. An alpha-level prototype with materials and tools will be completed by the first quarter of 1994, and a beta-level product will be built by the end of the year.

We will initiate an empirical study to validate the improvements experienced as organizations increase their maturity levels. Initially, business investment payoffs such as increased fidelity to schedules or reduced product defects will be reported in anecdotal case studies. As the time-series data on measures that we will begin to collect in 4Q93 become widely used, quantitative and rigorous reports of changes in process efficiency and product outcomes will be produced and released. Repositories of CMM-related data will be maintained for analysis during this strategic period. State-of-the-practice reports summarizing the benefits of software process improvement will be issued, as well as reports resulting from the data acquired from the SPAs.

Process Value Method (PVM) (1995). The PVM is a new initiative for the SEI. A preliminary version of the PVM is intended to be available for pilot testing late in 1994 or early 1995. The method would allow a project manager or SEPG to provide a perceived value from a business perspective for a process change whether the change is at the level of an SPA or SCE, a KPA, or a particular activity within a KPA. For example, if an organization is presently not performing inspections or peer reviews, then an SEPG could determine the costs, schedule, and quality effects that the organization will probably incur using the PVM. The method should allow the organization to understand what will be required in resources to make the change to perform inspections or peer reviews, and finally the expected results or values. The PVM is a repeatable method and thought process to guide the project manager or SEPG to estimate today's values and the expected new values after a process change has been implemented. The method will support both quantitative and qualitative aspects to determine value.

These defined potential products are reflected in the roadmap in Figure 3-4.

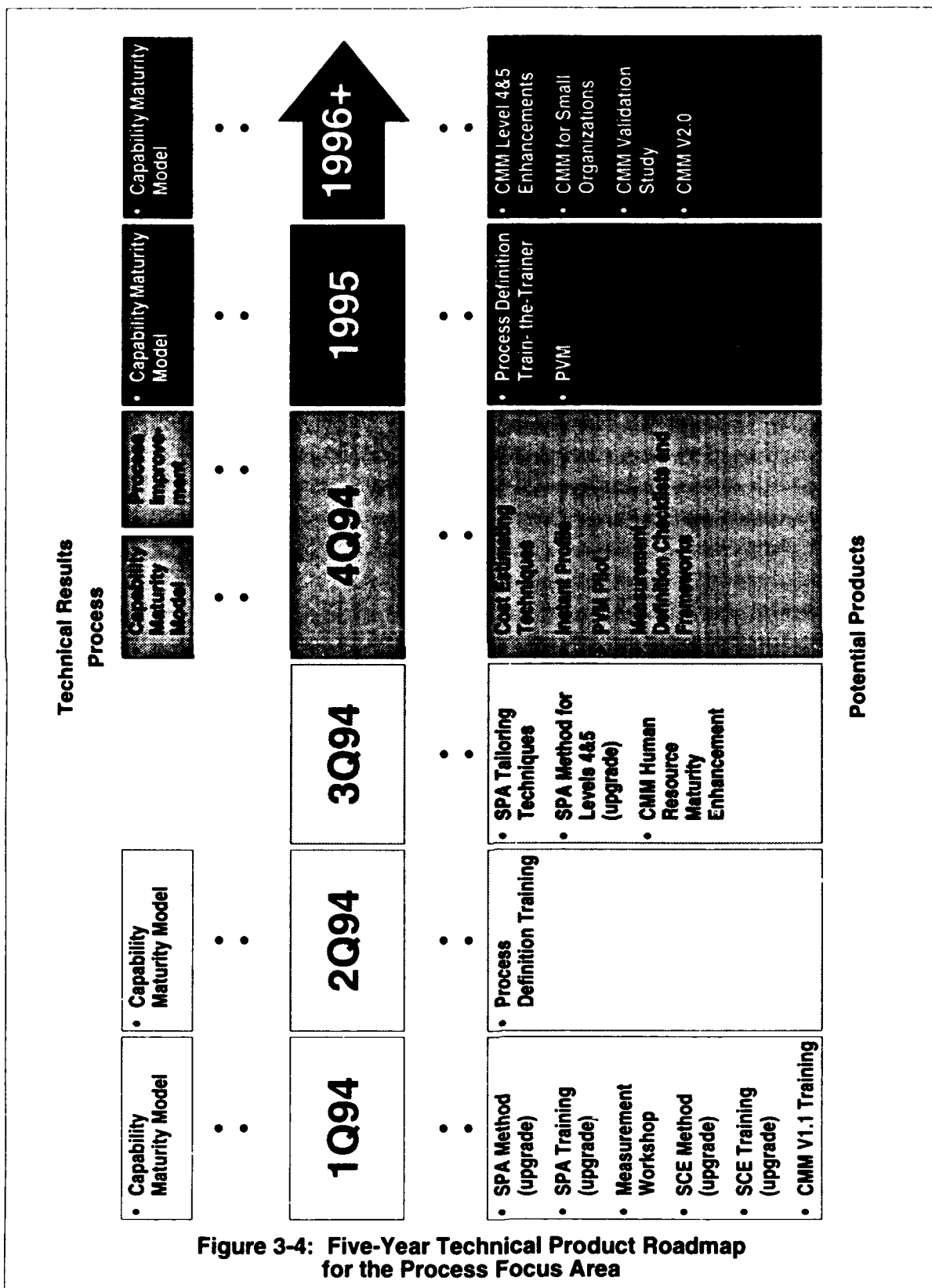


Figure 3-4: Five-Year Technical Product Roadmap for the Process Focus Area

3.2.2 One-Year Plan

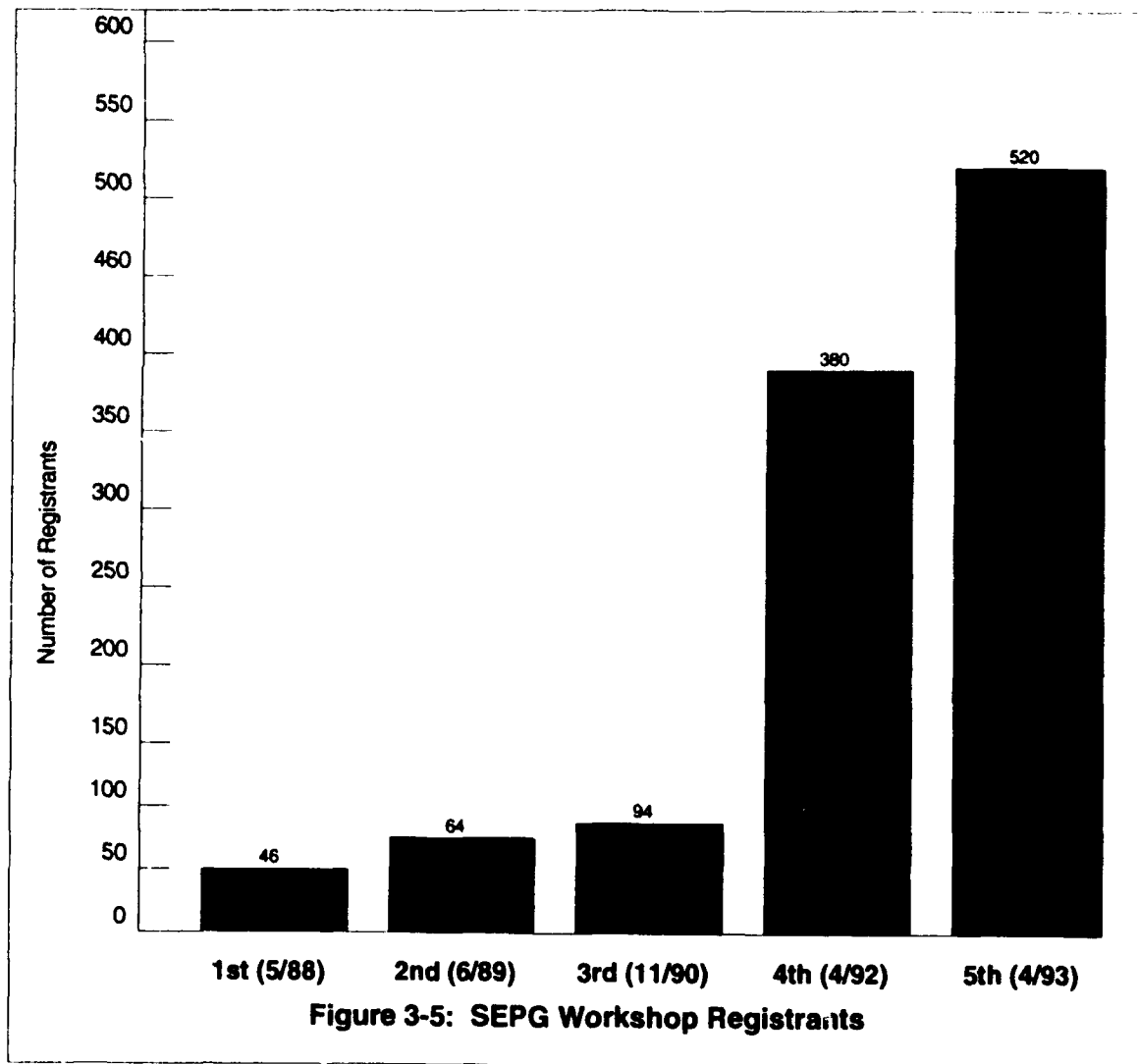
3.2.2.1 Context

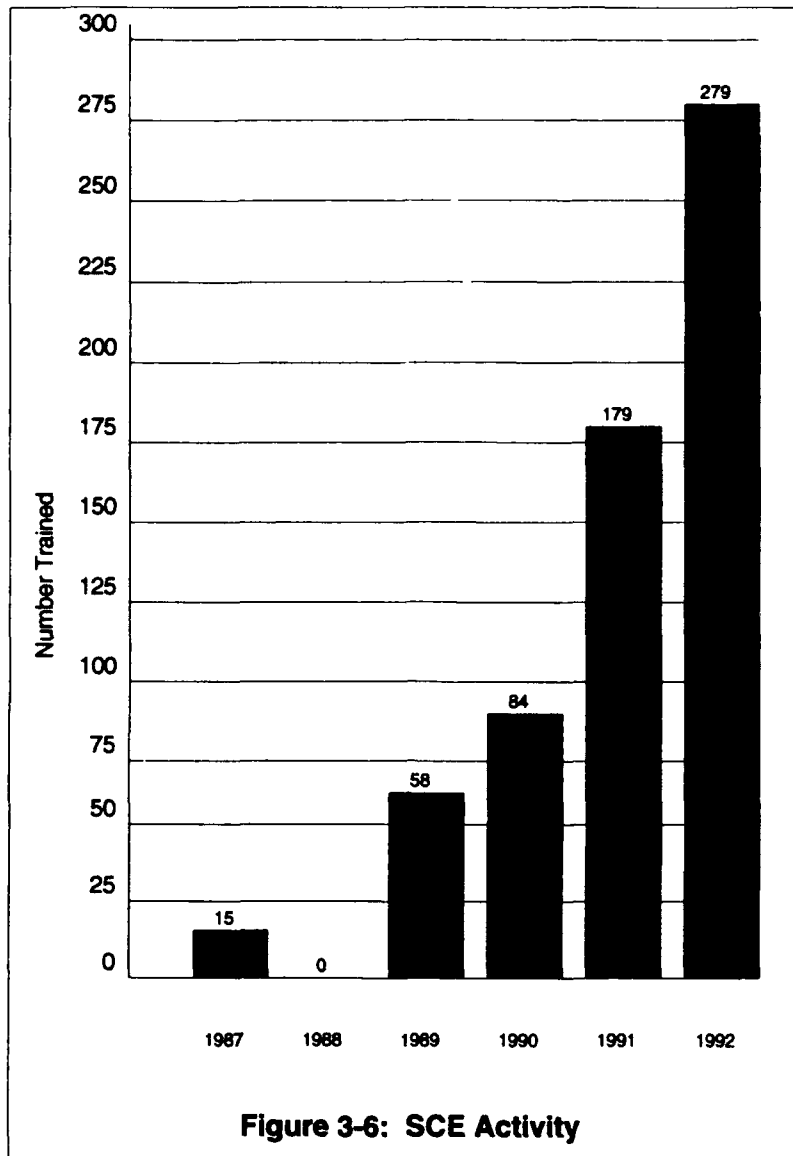
During 1994 we will complete the distribution of materials, methods, and courses based on CMM v1.1. The CMM v1.1 was distributed in 1Q93. Upgraded training and materials for performing process assessments and capability evaluations as well as the CMM will be maintained, updated, and reissued as necessary. The 1993 upgrades involved pilot testing of all methods, materials, and courses. Knowledge from these pilots and ongoing use will be integrated into the maintenance upgrades. We will complete the transfer of these methods on a limited basis to appropriate distribution partners (SPA Vendor Association, DSMC, etc.) to expand the availability of these methods to their relevant communities.

We will continue to expand the methods that are available to the SEPGs for process improvement by developing training courses for using process definition techniques and installing software measurement programs. We will continue to pilot these and other techniques with our strategic partners to get feedback on how best to design, teach, and package these techniques for broader distribution. These pilot programs are designed to enhance significantly the capability of military and other government agency organizations to develop and maintain software. We will be collecting lessons learned from these efforts and will be codifying them for dissemination to the broader software process community. We will continue efforts in support of the software cost estimation improvement initiative and focus on developing defined processes for cost and schedule estimation. Efforts to promote and influence standards activities will continue for the CMM and the software metrics standardization activities.

We will actively increase our involvement with resident affiliates during 1994. This skill base has helped process area initiatives in the past.

We will foster the development of a software process improvement community through the annual SEPG National Meeting and a national infrastructure of SPINs. In 1992 there were two SPIN groups in the United States. By 1993 there were five. It is anticipated that these groups will continue to increase. The SPIN groups represent the software organizations in industry in areas such as Boston and Washington, D.C. The 1st SEPG Annual Meeting had an attendance of 46 in 1988. In 1993 attendance was 520 (see Figure 3-5). Figure 3-6 shows the increase in training for SCE. These figures reflect the rapidly growing interest in process products and services in the software community.





3.2.2.2 Core Funded Activities

Core funding in 1994 will be used to complete a number of 1993 initiatives and to maintain and evolve the CMM. Some core funding will be applied to SPA and SCE for infrastructure, program management, guidance and advice, and partial maintenance of the products delivered through 1993. Additionally, we will apply core funding to begin the PVM, develop the CMM for small organizations, and explore the expansion of practices for levels 4 and 5 in 1994. In 1994, research in the process focus area will be conducted in the following areas as described in Section 3.2.1:

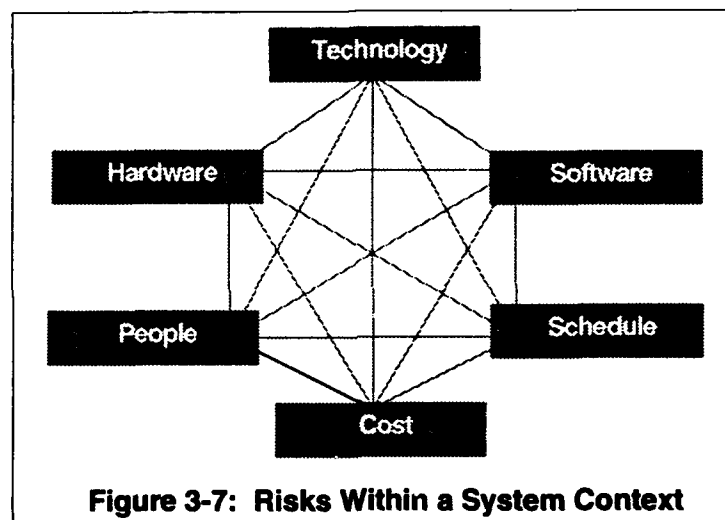
- software measurement
- software process definition
- CMM validation

3.2.2.3 TO&P Funded Activities

To continue to meet the growing needs of the software process community, the SEI must significantly increase its TO&P funding for the process focus area beginning in 1994 and continuing through 1999. Since a number of CMM v1.1 related products and services will be available, it is the intent to work with funding agencies in the government sector and with industry to directly transfer these products and services. We expect additionally to attract both government and civilian TO&P and cost recovery funds to help expand existing SEI products to meet the expected growth in user requirements for change and modifications that may become evident with broader use of the methods, materials, and courses. TO&P funding will be required to cover work with the human-resource aspects of the CMM and to make more specific the needs of clients of the PVM.

3.3 Risk

The software-producing community often makes technical decisions without adequately understanding the impact of those decisions because of software and system complexity. Management can also make decisions that drive inappropriate technical decisions, and then they are routinely surprised by unexpected technical problems, which usually manifest themselves in cost overruns and schedule delays. However, cost overruns and schedule delays are only symptoms of the root problems. Changing requirements, ad hoc development processes, new technology, and unprecedented engineering of technology, for example, represent sources of uncertainty and therefore risks to developing software systems. Not only is software critical to systems, but all areas in systems development are potential sources of software risks; therefore, risks must be viewed within a system context (see Figure 3-7).

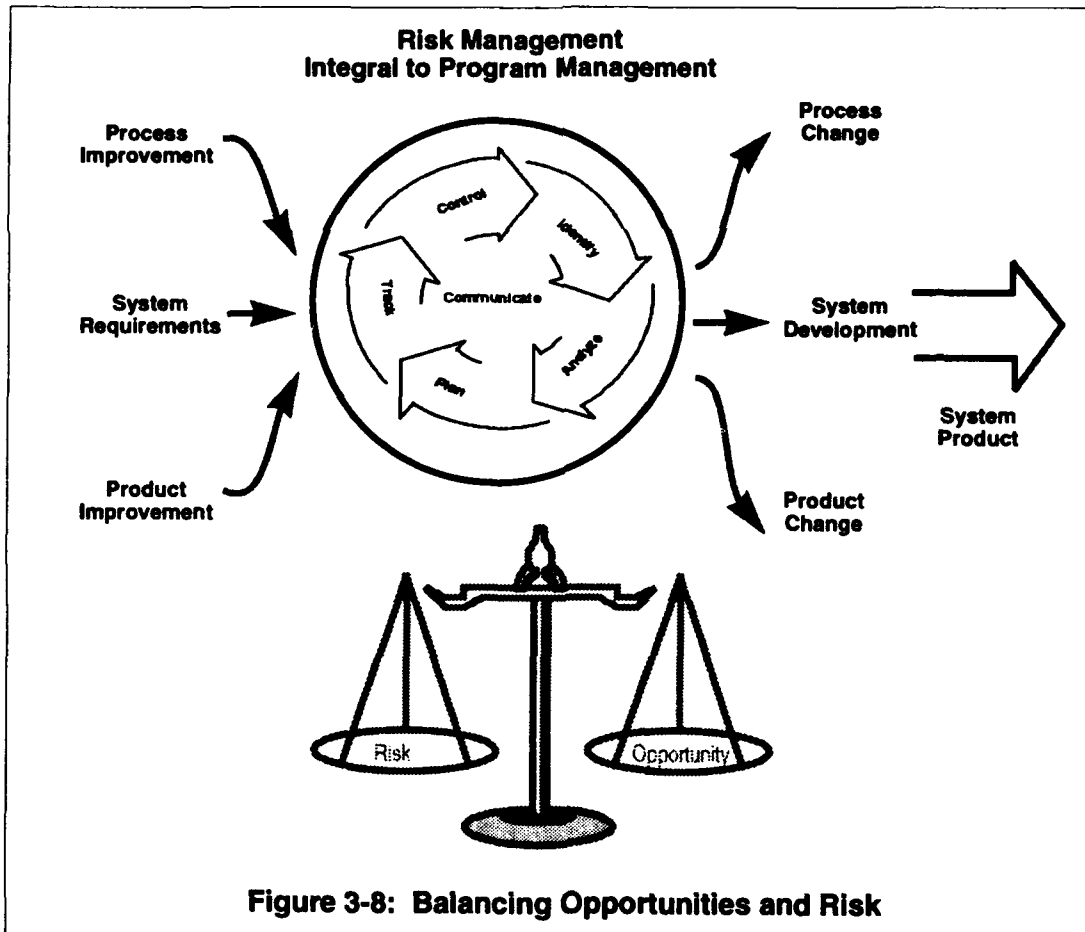


Validated and systematic methods and processes to identify and manage software technical risk are necessary if software development is to become an engineering profession. Engineering is structured and disciplined, even when dealing with problems. Software development must use structure and discipline and become more proactive and disciplined, and less reactive in dealing with software technical uncertainty. A systematic and validated process to identify and analyze technical uncertainty is necessary to move the software engineering community from one with ad hoc practices to one that manages software technical risk. Successful programs identify risks early, and, therefore, are able to seize opportunities such as exploiting technology with greater reward or lower risk.

Proactively addressing uncertainty is a rational approach in developing and implementing sound program strategies—strategies that weigh program opportunities and risks from both business and technical perspectives. For example, one would develop a program strategy on reengineering product XYZ by weighing the advantages of the technology and processes with

the uncertainties perceived from the existing and predicted conditions. A risk-aware approach supports any environment where change, precedence, or uncertainty is a factor. Reengineering, reuse, large-scale development, and unprecedented technology are a few examples where risk management would be effective in the acquisition and development strategy as well as in the program implementation (see Figure 3-8).

The SEI is focused on identifying and analyzing software technical uncertainty to present the decision maker with the right information in a timely fashion. Since such generally accepted methods do not exist currently for software, we seek to develop processes and supporting methods that will systematically identify and analyze software technical uncertainty. We seek to improve program successes by identifying key risk management practices and the criteria by which they can be integrated into software development processes and program management. Risk management will be established as an organizational capability, executed on a continuous basis, and integrated within the context of program management.



3.3.1 Five-Year Plan

3.3.1.1 Goals

The goal of the risk area is to raise the maturity of the practice and management of software engineering through the development and transition of products and services, that enable software engineers and managers to better identify, analyze, and manage the technical uncertainties in software-intensive systems. This improvement will be achieved by providing proven and tested methods and processes in the form of education, training, questionnaires, and data gathered from government and civilian practices. Software engineers and managers will be able to make better decisions about the technical issues inherent in the next generation of software systems because they will:

- identify risks before they become problems
- communicate risks in a positive, non-threatening way
- resolve technical risks in a cost-effective way

In last year's plan, the goals in the risk area were conservative:

- Navy Program Executive Office (A) program managers will have incorporated software risk management into their acquisition process.
- DSMC and commercial training organizations will have continuing education offerings on software risk management.
- Major DoD contractors will have incorporated risk management into their software development process.

The SEI has made significant progress in developing a model of software technical risk management and validating those methods through its field activities. This improved understanding is reflected in more aggressive goals and product development.

Very little data exist today upon which to base quantitative measures of success for the SEI risk identification and management efforts. By 1999 we seek to have in place a system that will allow quantification and tracking of SEI effectiveness in reducing risk, and evidence that our techniques are effective. In the meantime, we are using the following as measures of success:

- Evaluations provided annually by our TO&P customers via "sponsor feedback forms."
- Numbers of clients who collaborate with us in independent risk assessments and team risk management activities.
- Risk identification training courses.
- The number of participants at the annual SEI Risk Conference.

We expect that by 1999, risk management will be using validated methods systematically in acquisition and development throughout the life cycle, and a new paradigm of team risk management will join the customer and developer in a cooperative spirit of proactive risk management. In addition:

1. Software risk management will be recognized as a key practice in the acquisition life cycle.
 - Risk management will be established in government and civilian standards with documented methods and processes.
 - System acquisition strategies and decisions will be based on information from proactive, systematic, and validated risk management methods.
 - Formal program reviews will be conducted using systematic and validated risk evaluation methods.
 - The DSMC will have courses on software risk management in acquisition.
2. Software risk management will be recognized as a key practice in the development of software-intensive systems.
 - Major DoD contractors will have incorporated risk management into their software development process.
 - The SEI will have an established national repository of risks and supporting risk reduction strategies based on information gathered from strategic partners and user networks.
 - Commercial training organizations will have continuing education offerings on software risk management.
3. SEI team risk management that builds on the strength of integrated customer and developer risk management will be adopted as the benchmark for integrated product teams using systematic and validated methods and processes.
 - Major government programs will have integrated customer and developer risk management activities—team risk management.
 - Team risk management will have integrated technical, cost, and schedule risk management into continuous, routine program management in major programs.
 - The DoD will have adopted team risk management as their acquisition and development practice.

3.3.1.2 Strategy

Our strategy is based on an evolutionary approach that builds credibility and awareness through contact with actual projects and hands-on application. We nurture strategic partnerships to establish an operating base to leverage experience and knowledge through testing of methods. For example, we have conducted numerous interviews and assessments with civilian and government programs to identify the needs and current practice of software risk identification and management. This information has been used to guide our development effort.

Through interviews, assessments, and collaboration with strategic partners, we are building a knowledge base about software technical risks, and strategies for dealing with those risks. The SEI is in a unique position to gather this information from many sources and aggregate it for use by the community. These data (our knowledge base) provide a foundation for the improvement of identification methods and a source of information that developers can use to identify and manage software risks on their projects. Workshops, working groups, seminars, and conferences will also be used to gather information, obtain feedback, and raise the awareness level of the community. This interaction with the community also establishes the baseline against which to measure progress in improving the practice of software risk management.

In earlier work, we developed a paradigm for risk management (see Figure 3-9). The paradigm is our model of how the different elements of software risk management, described below, interact:

- **Identification** locates risks before they become problems and have an adverse effect on a program.
- **Analysis** converts raw risk data into decision information.
- **Planning** turns the information into decisions and actions both present and future.
- **Tracking** monitors the status of risks and actions taken against risks.
- **Controlling** corrects for deviation from planned risk action.
- Finally, **communication** provides the visibility and feedback on risk activities, current risks, and emerging risks internally and externally.

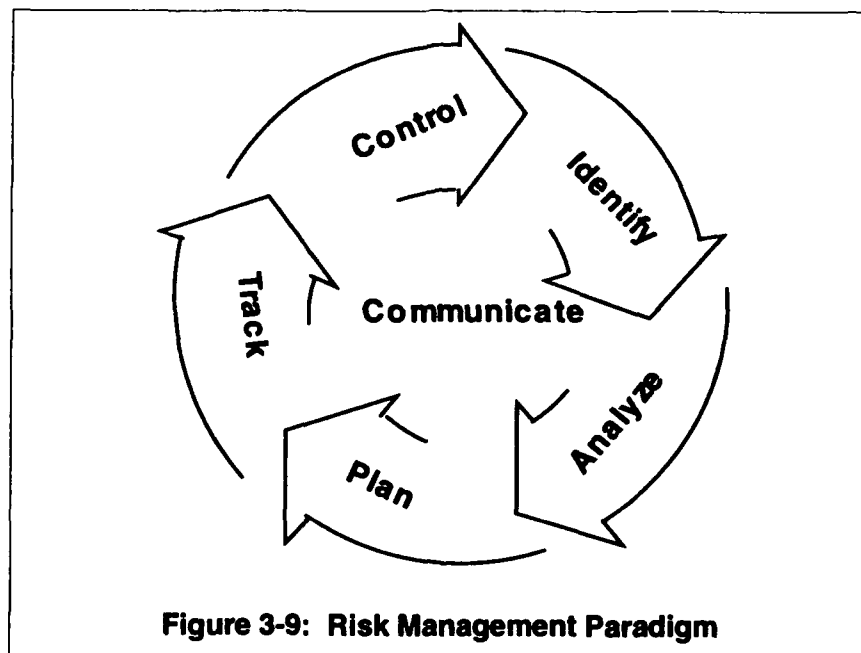


Figure 3-9: Risk Management Paradigm

The paradigm serves as a framework for describing how software risk management can be implemented. It provides the infrastructure for evolving the risk management building blocks and for integrating risk management into existing and future software development programs.

Our strategy is to elaborate and describe this paradigm as we gain experience from testing our methods. For each element in the paradigm, we plan to provide guidance on specific implementation methods and techniques. These methods will come from existing best practices or will be developed specifically to support risk management. Using the paradigm as a framework, we will solidify components of risk management methods and techniques into processes that can be put into routine, continuous practice.

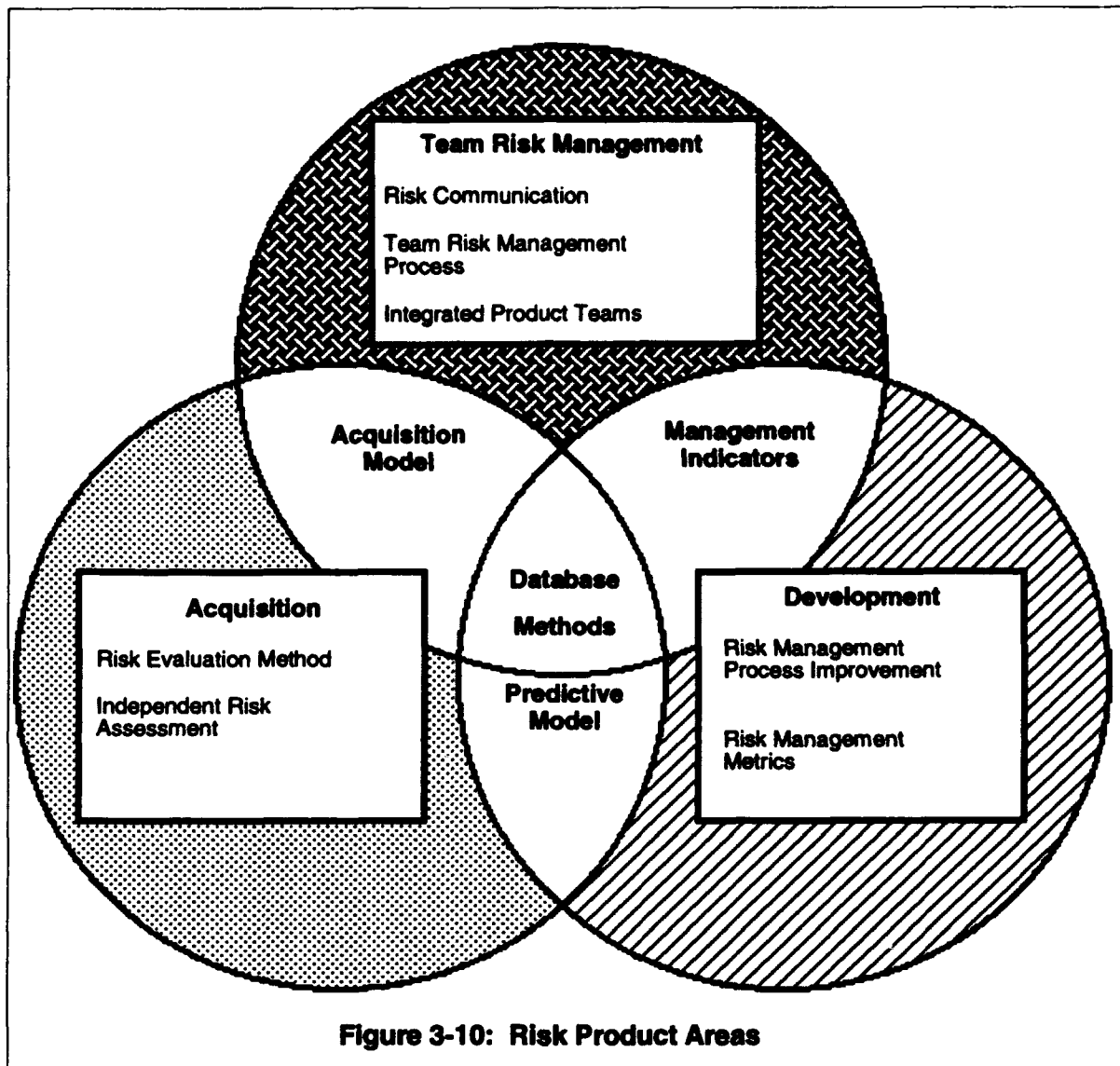
We are developing techniques such as a taxonomy-based questionnaire and a matrix correlation method to discover risks. By applying these systematic and disciplined methods to several projects through field activities, we are also gathering a valuable database of information about risks and actions to resolve risks on a wide range of projects. These data are adding to our knowledge base to be shared on a wider scale as a source for managing risks.

Risk planning addresses strategies to mitigate risks. Methods to address planning are evolving as we work with our strategic partners. The track and control elements are an integral part of project management, and, much like planning, we have developed a simple approach that we are evolving with our strategic partners. The many existing practices in project management provide the basis for integrating risk management into existing practice.

To be effective, we must not only provide methods for identifying and managing technical uncertainty, but we must also provide methods to facilitate the communication of risk issues. In addition to the usual cultural barriers, the common negative perception of risk makes change even more difficult. The communication process must depersonalize risks so they are viewed as opportunities for program success. Factors for communication are in all the method development and field testing activities.

From the perspective of transition to the client community, SEI risk work focuses on three areas: acquisition, development, and team risk management. The risk management methods discussed previously are enhanced and integrated for effective application in each area. (See Figure 3-10.) All application areas contribute to and use a common repository of risks and mitigation strategies (see page 53).

In the acquisition area, the focus is on evaluating the software technical risks for a specific program either by a customer or independent agent. The potential acquisition products cover risk evaluation and independent risk assessment methods and their application within the acquisition community. Future effort would evolve risk identification and analysis methods as major contributors for determining an acquisition or development strategy such as a specific reengineering approach on a given program.



Risk management improvement is the main focus in the software development area. The emphasis is on evolving potential products to address improving the organization's risk management practices of organizations. This will include defining key practices as an integral part of the SEI CMM. Several companies are participating in the development and testing of risk management methods through technical collaboration agreements. These agreements allow the SEI methods to be proven in field conditions and the best practices to be shared by the entire software development community. This work is a vital part of the transition strategy for rapidly improving the state of the practice of risk management.

The team risk management area promotes a new paradigm of shared commitment to manage program risks as a team on a continuous basis. The potential products are an elaboration of the team risk management concept and a user's guide that promotes risk management as a continuous process integrated into program management practices. Specific methods are developed addressing joint activities and cooperative risk management in the context of the risk paradigm. The team risk management process is developed through multiple strategic partnerships allowing long-term, joint collaboration with the customer and developer on active programs. This integrated product team work is a vital part of the SEI transition strategy for rapidly improving the state of the practice of risk management. Since communication is vital to a team approach, other potential products will include methods that enhance inter-communication and lower the inherent barriers to communication.

To ensure a successful transition strategy, we are approaching transition systematically by targeting products to leverage limited SEI resources and to support adopting and sustaining the technologies. This includes community awareness activities such as conducting the annual SEI Software Risk Conference, and presenting risk management tutorials at conferences and SEPGs, and to customer and developer working groups. These activities provide the additional benefit of feedback on our work. Education, training, collaboration, and publications will be our primary instruments for affecting understanding. For example, education and training includes the risk taxonomy questionnaire tutorial and software risk management tutorials. Collaboration is addressed by the field activities to test specific methods with our government and industry partnerships. Installation is affected by teaching executive courses on risk management and conducting independent risk assessments. Finally, we are addressing adoption and institutionalization by conducting team risk management, risk management improvement, and independent risk assessment activities, and by producing guidebooks and handbooks such as the Software Risk Capability Improvement Guide, Team Risk Management Handbook, and a software risk reference or textbook.

3.3.1.3 Potential Products

Global

SEI Software Risk Conference (annual). This conference addresses the wide range of needs of SEI customers, from practitioners to managers in both the government and civilian sectors. The purpose is to provide a forum for the exchange of ideas, an opportunity to be exposed to best practice, and an awareness of current experiences in software risk management. The conference provides current information on risk management methods, theory, and practice through invited presentations, panel discussions, and workshop formats. It has proven to be an important mechanism in establishing a community of research and practice in software risk management.

Risk Data Repository (1996). The risk data repository will be populated initially with data collected from field tests and risk assessments conducted by the SEI and strategic partners. The database will contain entries in risk and mitigation strategies and will be designed to use more data as they become available. As a repository it would provide reliable information on what risks programs have faced for particular situations and over their lifetime, and how they dealt effectively with those risks. The repository will provide a two-way avenue of information to clients and will become more robust over time as new information is received and validated.

Development

Risk Identification Training Course (4Q 1994). This training module will prepare software knowledgeable engineers to identify software technical risks. The module will be designed to prepare a software risk assessment team; however, the instruction will be useful to technical managers interested in methods that systematically identify software technical risk. The participants will learn to use methods to identify software technical risks and how the methods are applied in a program-assisted risk assessment process. The module development includes instructional material, method descriptions, lectures, and active participation to learn the methods. The taxonomy and matrix risk identification methods and their application will be described in detail including examples from actual experiences and pitfalls to avoid.

Tailorable Taxonomy-Based Questionnaire (1995). As part of continually striving to make the risk identification process as practical and efficient as possible, a tailorable taxonomy-based questionnaire will be produced. This product will take into account the characteristics of projects being assessed including the domain, life-cycle phase, and type of project.

Risk Analysis Training Course (1995). The risk analysis module will address analysis of the risks identified by the taxonomy method, the matrix method, or any other method a developer uses to identify risks. The module will cover training the methods to produce a prioritized list of risks clustered by taxonomic categories for a project to plan mitigation strategies. Similar to the Risk Identification Training Course, it is designed to prepare a software risk assessment team, and it will be useful to technical managers as well.

Software Risk Capability Improvement Guide (1996). This guide will address practitioners and managers who need an understanding of the best practices in risk management. The guide will include an elaboration of the risk management paradigm and how risk management is made an integral part of project management. It will describe in detail methods and techniques that implement risk management as illustrated by the paradigm (for example, how to identify, analyze, and take action on software risk). It will provide a framework for integrating these methods as routine practice into the specific environment of the project. This guide will provide the supporting material for training courses and improvement projects on risk management.

Acquisition

Independent Risk Assessment (1Q 1994). This service applies the Software Risk Evaluation (SRE) method on a program at the request of a sponsor to produce findings, conclusions, and recommendations about software issues and risks. The SRE method provides a systematic and validated way to assess program risks. It is an attractive alternative to "red teams" because it is effective and repeatable. More importantly, the SEI systematic method tends to depersonalize the findings and defuse the adversarial barriers that mark "red teams." This service provides the client with issues and recommendations as well as opportunities where the SEI can apply its other products and services to improve a client's maturity in organizational and management processes, technology, and practitioner skills.

Software Risk Evaluation (SRE) Handbook (4Q 1994). This document will provide a full description, along with the appropriate tools, of how to conduct an SRE. It is intended for use by expert reviewers as a supplement to training, and as a reference to conducting software risk evaluations. Expert reviewers will be able to follow the steps within the handbook and refer to it for suggestions in identifying issues, drawing conclusions, and making recommendations.

Software Risk Evaluation (SRE) Train-the-Trainer (1995). This training will provide the necessary support for expanding the application of the SRE across the DoD. It will consist of two parts: (1) a training methodology and supporting documentation that the SEI will use to train selected DoD employees and through which they can become certified to instruct other DoD employees to perform a SRE, and (2) a training methodology and supporting documentation that the previously trained and certified DoD trainers will use to prepare other government employees to conduct SREs.

Predictive Decision Tool (1995). This tool is designed to provide the acquisition manager with a way to apply the resulting data from an SRE to predict possible outcomes. This data will be used to create predictive scenarios that can be managed by program management. The scenarios will also be clearly defined by means of the SRE-identified risks to the program, and the actions that must be taken to mitigate those risks. The tool will be designed to use data provided by other data gathering mechanisms such as the U.S. Army Readiness Growth Model.

Team Risk Management

Team Risk Management (1Q 1994). This is a service in which the SEI applies the team risk management paradigm to customer and developer strategic partners on selected programs. The SEI is applying methods that have expanded the concept of integrated product teams to cooperative risk management. As a result, both customer and developer (i.e., government and civilian) receive methods for independent and joint activities that engender cooperative risk management. The SEI installs the methods within the customer and developer organizations and facilitates the joint or "team" activities.

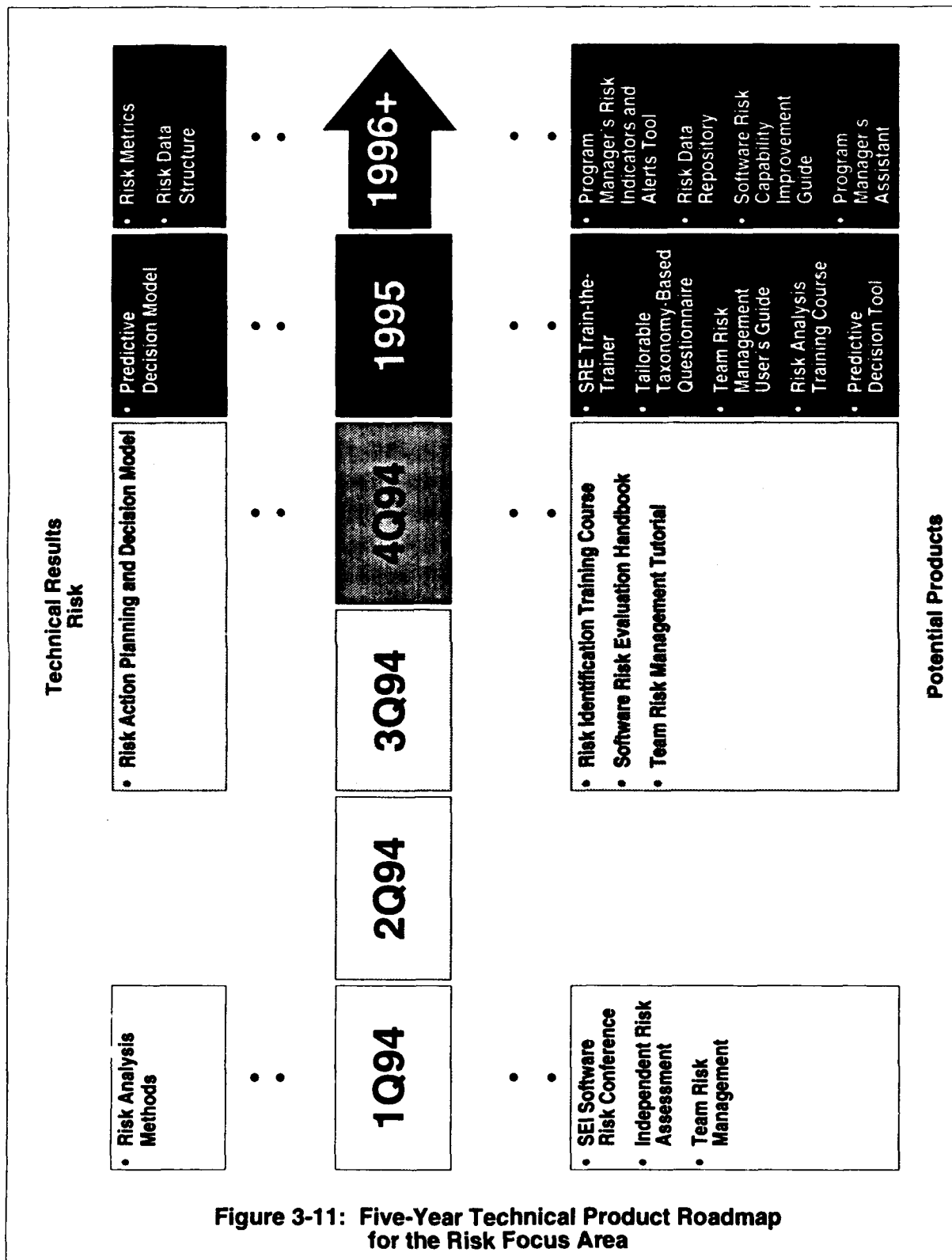
Team Risk Management Tutorial (4Q 1994). This tutorial will provide a training package to address both customer and developer participant instruction for team risk management. The training material will be designed to prepare the customer and developer to initiate and sustain independent and joint risk management activities on a continuous basis.

Team Risk Management User's Guide (1995). This guide will address customer and developer management processes and methods to implement and sustain continuous team risk management. The guide will elaborate methods and tools for individual and joint activities that integrate into a continuous program management infrastructure. The guide will describe how these become a part of the existing program management practice.

Program Manager's Risk Indicators and Alerts Tool (1996). This product provides management tools for risk identification and tracking. A major focus will be on making the program risk exposure visible, relevant, and understandable to managers. As mentioned previously, the instrumentation in the space shuttle or a nuclear power plant control room provides vital information on the health and status of systems. The SEI will produce mechanisms, e.g., a program manager's control room, that would provide minimum essential management indicators and alert information much like gauges and indicator lights to provide real-time information. By also correlating the situational data to the data in a repository of risk management, both common risks and mitigation strategies data, management would have a predictive capability to identify new risks and possible outcomes.

Program Manager's Assistant (1997). This longer range product will unify the methodology of risk management and the SEI risk data repository as a support tool for program managers. The computer-aided approach will provide assistance in identifying risks proactively by exploiting the SEI risk data repository to be alert to systemic or common risks. The tool will support analysis and tracking of risks on a continuous basis, building on the capability of the program manager's risk indicators and alert tool, and the predictive decision tool. It will also assist in developing mitigation strategies and exploiting data from the repository to suggest mitigation strategies to prevent risks.

Figure 3-11 shows the five-year technical product roadmap for the risk focus area.



3.3.2 One-Year Plan

3.3.2.1 Context

SEI risk activities focused initially on the creation of software risk management methods to identify and analyze software development risks. In 1993, the SEI put more emphasis on products to support acquisition. The Software Action Plan (SWAP) Working Group (WG) sponsored an SEI activity to develop a process of expert review to evaluate software risk in major acquisitions within the DoD. To accomplish this set of tasks, an initial deliverable was completed and delivered to the SWAP-WG on April 30, 1993. This deliverable, the SRE, Version 0.1, was designed around the taxonomy-based questionnaire (TBQ).

Work continues on the TBQ for the identification of development risks. The TBQ evolved through a research program drawing on software development expertise, the analysis of software development literature, and analysis of data collected through applying the questionnaire in a series of field tests on software-intensive projects within various government and civilian organizations. The questionnaire is based upon a taxonomy of software development focused on identifying risk throughout the development life cycle. The TBQ has proved to be an effective method not only for the identification of project risks, but also as a catalyst for communication within projects. The taxonomy-based questionnaire method was published by the SEI in June 1993 (technical report CMU/SEI-93-TR-6, ESC-TR-93-183).

We also developed and presented a one-day tutorial that provides insight to executives and middle management about software development risk and what it means to the success or failure of a software-intensive project to raise the awareness of the necessity of well-organized risk management to increase the probability of project success. The tutorial provides the foundation for additional details of identification and analysis training course modules being developed in 1994.

NOAA provided an opportunity to test the risk identification and analysis methods directly with a government program office, laying the groundwork for risk management within a large acquisition program office. This provided confirmation of activities that support a team risk management paradigm. This work will develop an action planning/decision model that will provide an approach to developing risk mitigation strategies. It will describe the criteria for developing action strategies and indicators and alerts to track changes in the risk state.

The Navy PEO(A) continued to provide the SEI with unique opportunities in 1993 to evaluate our approach in both government and civilian settings. This has allowed us to evolve and test our team risk management concept in active programs. As a result the team risk management methodology is documented and tested to support development of the team risk management tutorial and user's guide.

The ongoing collaboration with government and industry partners provided the opportunities to develop and test other methods and concepts such as risk analysis, action planning, and risk metrics.

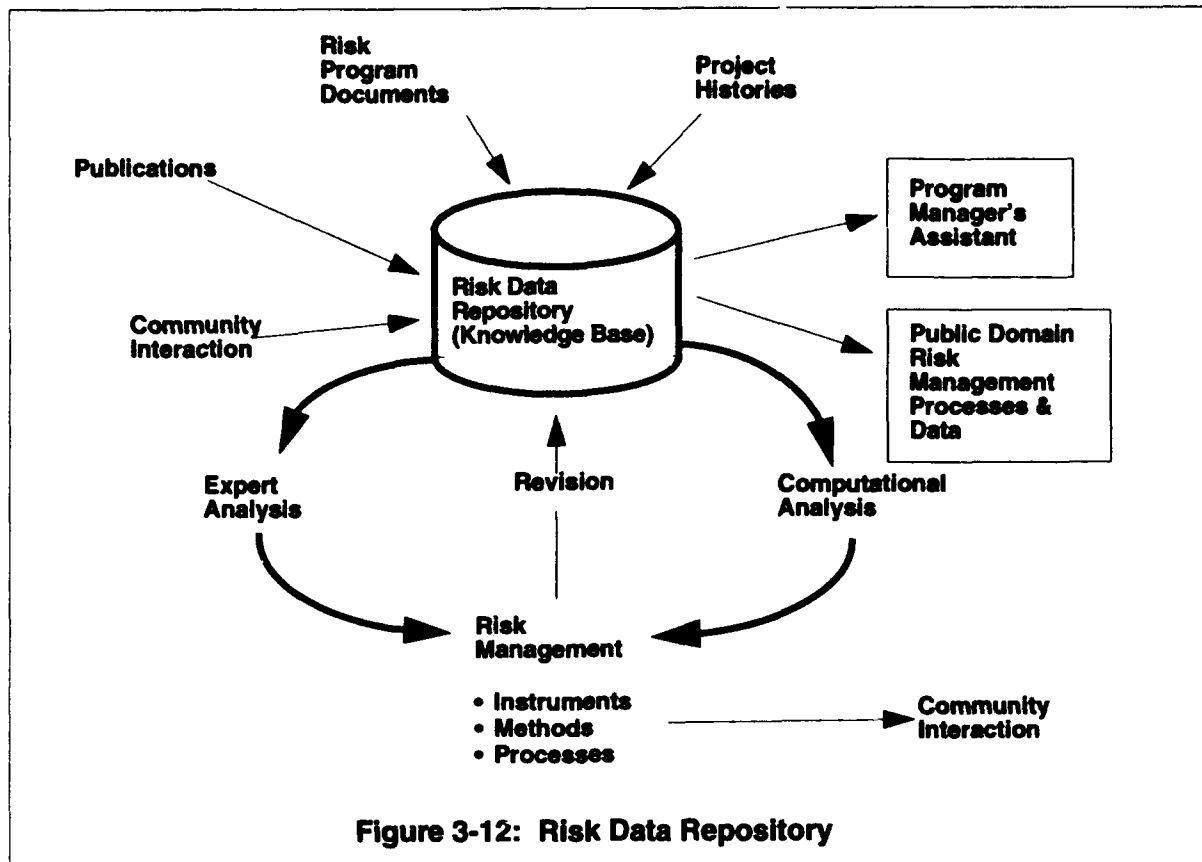
3.3.2.2 Core Funded Activities

The third annual Software Risk Conference will be held in Pittsburgh in March 1994. The conference will consist of refereed papers, panels, and invited speakers from industry, government, and academia and will help to raise the awareness level of the community pertaining to software technical risk and how it fits into software program management. We expect an even larger representation of information and experiences with risk management practice.

Several companies will continue participating in the development and testing of risk management methods through technical collaboration agreements. These agreements include tasks that the company and the SEI will further develop and evolve as risk management methods. Through these technical collaboration agreements not only are the methods developed by the SEI being proven in field conditions, but also the best practices of each company are being merged and integrated into a total risk management framework for the entire software development community. This work with companies is a vital part of the transition strategy for rapidly improving the state of the practice of risk management. The field work will provide the empirical data to improve and update the TBQ. Further development of the method will provide the capability to tailor the questionnaire to allow for differences in the life cycle and the product application or program domain.

We have already gathered a plethora of data from work with government and civilian sectors and are building a database for risk information. We plan to develop this work into a data repository. The risk data repository will become a community asset containing invaluable data to develop empirical and model-driven approaches to software engineering. One of the major problems facing software engineering is the lack of accessible data about the development and use of software products, what software development practices have been tried, and their effectiveness in particular contexts of use. Consequently, we are presently forced to resort to non-empirical arguments in deriving many software engineering methods, tools, and approaches.

The data in the repository will include common risks, risk mitigating actions, results, and lessons learned. Once obtained, structured, and analyzed, the data will also yield rich information on the relationships among risks, risk causes and attributes, and relative values of risks that will, in turn, be used to support the determination of risk ordering and prioritizing. Figure 3-12 is a schematic of how we envision repository use in risk management.



The central research issues in developing and populating the risk data repository are about structuring these data in a form that will enable access in developing and practicing risk management. Other supporting research issues are: (1) the determination of the types of data that are needed, (2) the identification of existing and potential sources of these data, (3) the methods to acquire existing data, and (4) the methods to capture data from potential sources (i.e., that are generated in the course of software development but pragmatically difficult to record). These issues will be addressed by continuing the current efforts in information analysis and modeling at the SEI and the CMU Engineering Design Research Center (EDRC).

The Risk Identification Training Course, targeted toward software-knowledgeable engineers to identify software technical risks, will evolve from our field testing in the third quarter of 1994. This material is building upon the previous work developing and testing the TBQ and brings additional methods for identifying risks.

Research will begin on a predictive decision model (PDM) to support software risk evaluation. This model is the foundation for a tool to provide the acquisition manager a way to apply the resulting data from an SRE. The predictive decision tool will be based upon this model, which makes use of various types of data and regression analysis to arrive at levels of impact from causative values of identified variables. The variables come from risk identification methods such as the TBQ. Applying weighting factors, the temporal site (time of data gathering) will specify the current status of the software development program being evaluated. Using rigorous statistical analyses, the predictive method will move the temporal site forward in time to predict future outcomes. The ability of the model and supporting tool to accomplish this prediction is dependent on the clear identification of a program's risks and the relative stability of the development program from major situational changes. Major change activity requires new risk identification data to refresh the prediction. By reviewing the predictions and coupling them with past situations, trend lines can be identified to assist the decision maker.

3.3.2.3 TO&P Funded Activities

The SEI will document the SRE method in the first quarter of 1994 to support the independent risk assessment service performed by the SEI. This service provides a client with issues and recommendations as well as opportunities where the SEI can apply its other products and services to improve the client's maturity in organizational and management processes, technology, and practitioner skills.

We are working with the Coast Guard Program Systems to Automate and Integrate Logistics (SAIL), using the SRE method. The thrust will be to evaluate the four competing architecture concepts for technical feasibility and risk. This provides the framework for future improvement needs that the SEI can apply from other products and services.

The SRE activity will initiate train-the-trainer product development and an SRE field operations handbook and test them during actual evaluations. The SRE Handbook will be completed in the third quarter of 1994, and the train-the-trainer product will evolve in 1995. These components are necessary to support the execution of an SRE, training the personnel resources that the DoD already applies to standard acquisition program evaluations, and reference material to support the conduct of an SRE. Work that began in 1993 to support independent risk assessments will be delivered as services to sponsoring agencies that fund these activities.

Direct work in programs through strategic partnerships is important for most development, but vital for developing the team risk management methods. We will continue collaboration and development focusing on integration of risk management into program management and developing techniques and guidelines to better communication. All the collaboration efforts will

add to our database of risks and mitigation strategies. Implementing team risk management processes will be provided as services to selected strategic partners that include both customer and developer. This process and the supporting methods will be documented by the first quarter of 1994.

The Navy PEO(A) strategic partnership will continue our collaboration on team risk management through 1994. We anticipate other long-term collaborations with other services and NOAA that will broaden our experience base and the opportunities to develop information about risks and mitigation strategies for our data repository. In the fourth quarter of 1994 we will provide what we have learned as a tutorial on team risk management. We will be focusing much more on additional risk analysis methods and developing risk mitigation strategies. We will also be laying the groundwork for risk indicators and metrics. These methods will all be information for a team risk management user's guide in the future.

Another important concept is the cooperative use of data to predict useful outcomes for team risk management. Much like the instrumentation in the space shuttle or a nuclear power plant control room that provides vital information on the health and status of systems, we are exploring what data can be used and how to use them to track and predict items at risk. These methods would include management indicators and alert information to provide instantaneous assessment of the situation to management. By also providing the correlation to the data in the previously mentioned repository of risk management data, the situational data would provide predictive capability to identify new risks and possible outcomes. The methods require both a sound technical foundation and effective communication attributes to become useful to managers; therefore, there is a focus on minimum essential information and its presentation.

Based on the results of our team risk management activity, we will begin to document information into a user's guide for team risk management. The guide will be completed in 1995. We will use to capture much of the information of a complete paradigm of cooperative risk management.

3.4 Methods and Tools

Demand for software intensive systems is not only constantly increasing, but the systems are growing in both size and complexity. In the past, systems were built in an ad hoc fashion, lacking a systematic approach. Engineering knowledge about a system was rarely preserved other than through the experience of individuals. Larger systems were created making little use of existing components. When existing components were used, they generally required significant work because they were never designed to interoperate. Current software engineering practices cannot keep up with the demand for engineering software systems as well as reengineering the legacy of old systems.

Improvement of the current state will be accomplished through codification and systematic use of engineering knowledge rather than looking for the best method or tool to perform an engineering function. Certainly that knowledge can be more efficiently supported by appropriate methods and tools, but more effective and efficient software engineering practices have to be employed to cope with this increased demand. Other engineering disciplines have addressed this problem by building a body of engineering knowledge in the form of product models and using them systematically. The desired state is for software engineering to become an engineering discipline that engineers and reengineers systems effectively and efficiently by:

- The use of system models
- Codification and maintenance of engineering information and experience
- Automation through computer-based support

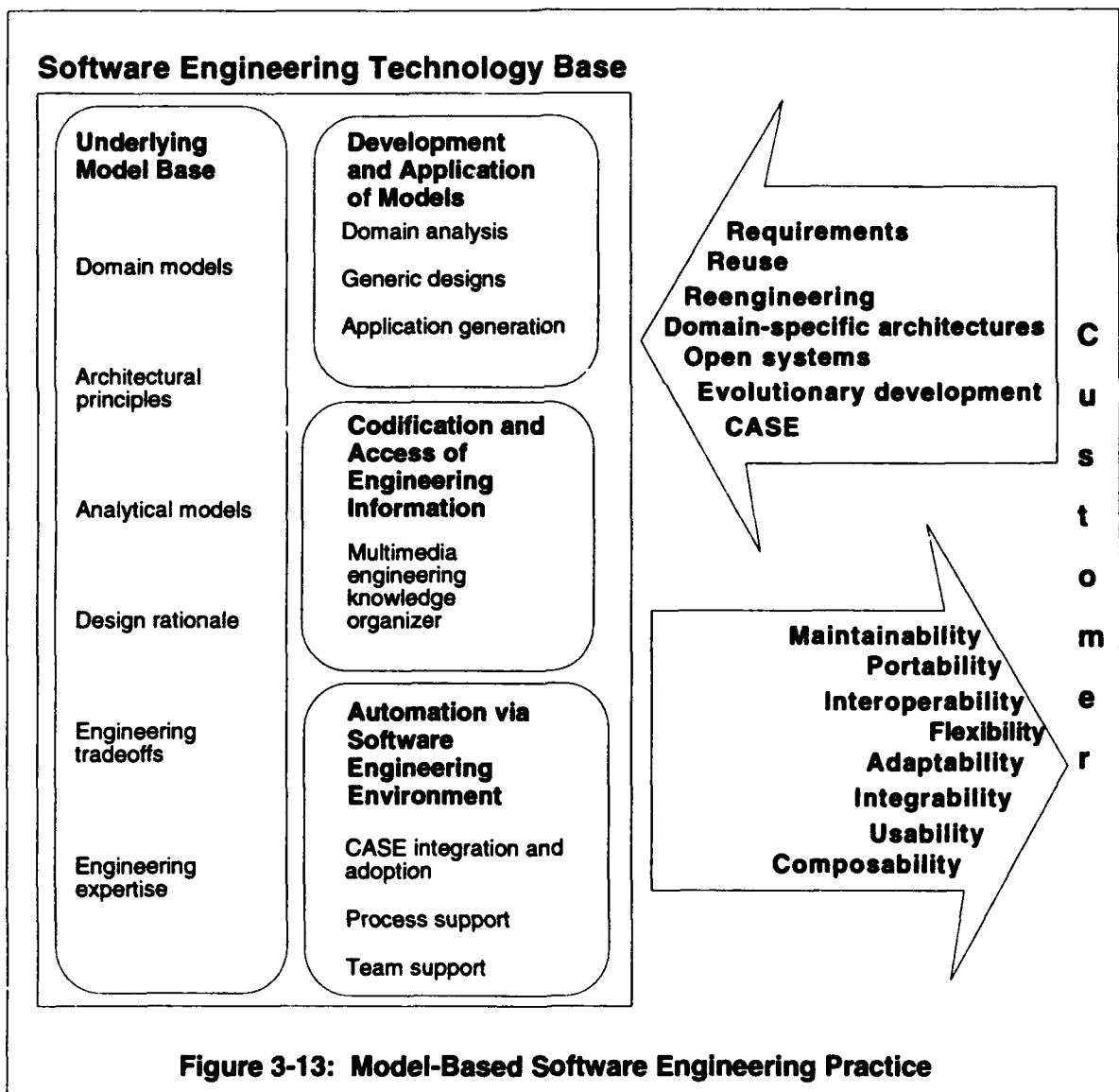
We refer to this practice as model-based software engineering (MBSE). Figure 3-13 illustrates the vision of an MBSE practice.

Models represent views of systems, capturing relevant aspects from a particular perspective. Examples of models include domain and architectural models as well as timing and performance models. As models become more formal, the approaches become more analytical and quantitative. Application engineering based on models leads to reuse of existing artifacts.

Engineering knowledge concerning a system under development has two components. The first focuses on capture, representation, and access to system understanding through engineering information recorded in a system design record. The second component focuses on engineering experience by capturing, representing, and offering expertise in the form of rationale, tradeoffs, and design decisions. These experience modules complement the information already encoded in the models. The increased amount of available engineering information may result in information overload, making it critical to provide intelligent access.

Computer-based support can provide automated support for the MBSE practice in a number of ways:

- Software engineering environments (SEEs) that provide team support and support for software process execution. SEEs are large systems that need to be engineered and whose proper adoption is critical to improvements in efficiency.
- Application generation from high-level description.
- Intelligent software engineering assistance through active electronic handbooks.



MBSE is a strategic technology approach to improve the engineering of software systems. It can address a number of engineering problems that are reflected in and are the root of customer interests. Engineering problems are expressed in terms of a number of "-ilities," as shown in Figure 3-13. Each of the engineering problems can be addressed through a mapping of the problem to the elements of the MBSE technology strategy. This mapping consists of a technical solution strategy as represented by MBSE and real-time distributed systems technologies, as well as risk analysis, engineering process, product and process measures, and improvement programs. Current customer interests include requirements engineering, reuse, reengineering, open systems, evolutionary development, and computer-aided software engineering (CASE).

3.4.1 Five-Year Plan

3.4.1.1 Goals

The mission of the methods and tools focus area is to provide technical leadership in improving the effectiveness and efficiency in engineering and reengineering of software-intensive systems through increased systematic application of models supported by methods and tools. The goal is for defense and civilian contractors and agencies to improve their ability to engineer and reengineer software-intensive systems through an MBSE technology solution strategy. The SEI will be instrumental in making this happen in critical application domains such as command and control, simulation and modeling systems, embedded systems, information systems, and manufacturing. This translates into the need for a technology base and engineering approaches for applying it to customer engineering problems.

Success will be measured by the improvements defense and civilian contractors and agencies experience in engineering and reengineering software-intensive systems through an MBSE technology solution strategy.

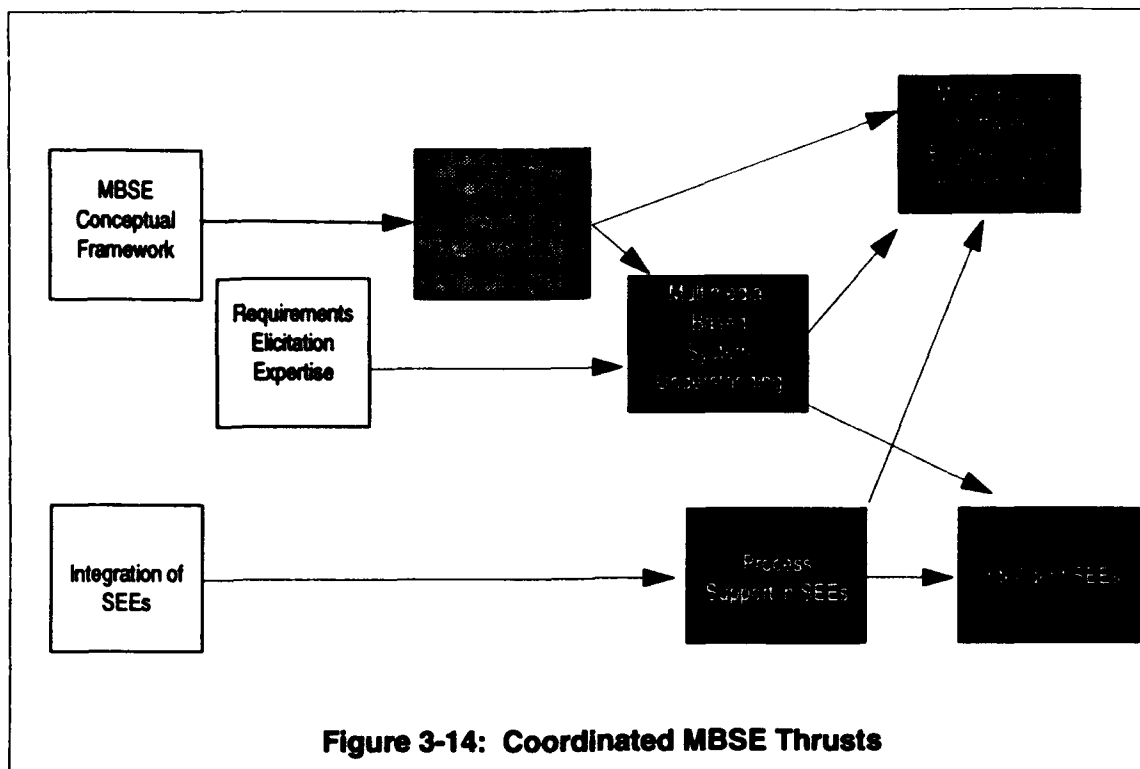
For example, by 1999 the model base should consist of domain and architectural models as well as domain-specific architectures that have been deployed across several application areas, including flight simulation, movement control, and command and control. It is expected that the software engineering community will routinely use these models in engineering and reengineering application systems. It is also expected that industry will have ongoing efforts to evolve the model base (reuse thrust) and that the models will increase in quantitative and analytical character.

Also, by 1999 a series of experience modules regarding critical aspects of software engineering should exist and be used routinely by practitioners. These experience modules will codify engineering knowledge (rationale and tradeoffs) to complement the information encoded in models. The modules as well as engineering information representing system understanding regarding specific application systems should be accessible in an intelligent, interactive multimedia (I^2M^2) environment to compensate for potential information overload.

Also, by 1999 flexible SEEs that support specific (model-based) software engineering practices should be routinely engineered and adapted by commercial systems houses to meet the needs of customers. This will require increased community consensus and standardization to improve more flexible SEE composition from different component suppliers. It is also expected that those who develop software-intensive systems will more systematically adopt CASE technology, resulting in increased effective use by software engineers. This is expected to improve efficiency. The SEI should be receiving consistently favorable reports from TO&P customers involved in activities leading to this model base and supporting SEEs. Interest in MBSE technology as expressed by attendance at the annual SEI Software Engineering Techniques Workshop and requests for the guides and other products developed in support of MBSE will serve as interim measures of success until the vision outlined above can be realized.

3.4.1.2 Strategy

The technical strategy is to evolve to a MBSE practice by building on efforts to focus, synergize, and leverage ongoing work in the methods and tools focus area. We will evolve and refine a conceptual framework for MBSE that incrementally integrates and fuses the three thrusts of the model base and its application, engineering knowledge codification, and engineering and adoption of SEEs into a cohesive and consistent improved software engineering practice. Figure 3-14 indicates how the thrusts build on each other's results.



The need for the SEI to take a leadership role and leverage its limited resources characterizes the strategy. We develop conceptual frameworks that become analytical tools for assessment of best practice and analysis of technology trends. The results and insights provide the foundation for practitioners, managers, and educators in the form of strategic guides to best software engineering practice, and to technology investors in the form of technology investment strategies to improve best practice. We leverage our limited resources by:

- Drawing from the experience of expert practitioners through case studies.
- Validating improvements in best practice through pilot projects with strategic partners.
- Advancing insights into improving best practice with resident affiliates on staff at the SEI and through cooperative projects with non-resident affiliates.
- Influencing the community through technical leadership in defense and civilian forums to build consensus in a technology area.
- Using these established forums and strategic partners as transition channels.

Efforts in this focus area depend on core resources as well as TO&P funding of up to 50 percent of the budget to successfully accomplish the outlined plans. TO&P funds that primarily have the character of technology development, case studies, and pilot application enable us to accomplish our goals.

Development and Application of Models

Evolution of the model base builds on work in domain modeling, software architecture design principles, and composition of systems via the application of structural models. It is expanded through population with models and refinement of the set of modeling concepts. Models in this context are views of application systems to be built. In its leadership role, the SEI provides a MBSE framework as the conceptual glue for relating modeling concepts and models. This framework accommodates assimilation of promising methods from the community and their integration into a cohesive MBSE process. Figure 3-15 illustrates such a software engineering process. The conceptual framework is extensible along two dimensions: addition of models, and introduction of new modeling concepts. Techniques for building product models allow others to populate the model base. The range of modeling concepts will be refined over time to incorporate quality attributes through cooperation with the SEI real-time distributed systems activities, the research community, and experience of expert software engineers. Fostering the application of models and domain-specific architectures across application areas will lead to better understanding of the leverage gained from a model base. The MBSE process is evolved to span the engineering life cycle; to incorporate system composition and integration, and evolutionary and incremental development. In this thrust resident affiliates and visiting scientists are important contributors. Collaboration with sponsors includes:

- Pilot work in the domain of movement control with the Communications-Electronics Command (CECOM)
- Exploration of reuse and domain analysis issues with the National Institute of Standards and Technology (NIST)
- Work on simulation through strategic partnerships with the Joint Modeling and Simulation System (J-MASS) and Navy programs
- Work on evolving the model base with industrial partners and with government programs such as Strategic Defense Initiative Office (SDIO) / Ballistic Missile Defense (BMD), Software Technology for Adaptable, Reliable Systems (STARS), Domain-Specific Software Architecture (DSSA), Central Archive for Reusable Defense Software (CARDS), and PRISM (Portable Reusable Integrated Software Modules).

Codification and Access of Engineering Information

Codified engineering knowledge complements system information captured in models through codification of rationale, expertise, and other associated information. The strategy in this thrust is to investigate techniques and tools to improve the software engineer's ability to capture, represent, and access reusable software engineering information, knowledge, and models. The initial focus is on the improvement of requirements capture and analysis, expanding into information required in a design record to evolve and maintain system understanding. At the same time, through fusion and application of several multimedia technologies, the practicality of intelligent computer-assisted access and learning to engineering information and knowledge is being demonstrated. In the long run, codified engineering knowledge will grow into an MBSE handbook, while the multimedia-based intelligent access and learning technology will merge with process-centered SEEs supporting execution of software processes (see below). The SEI has undertaken the following activities in support of the gathering and representation of engineering information:

- Joint projects with industry partners such as Texas Instruments (TI) for the gathering of engineering knowledge.
- Membership in the CMU/WQED Communications Multimedia Consortium, which has WQED Communications, the CMU School of Computer Science, and Intel among its members.
- TO&P arrangements with the Naval Supply Systems Command (NAVSUP) in the manufacturing arena dealing with legacy engineering information.
- Leadership roles in relevant public forums, such as chairing an international workshop and conference dealing with multimedia and being a member of the Institute of Electrical and Electronic Engineers (IEEE) Task Force on Multimedia Computing.

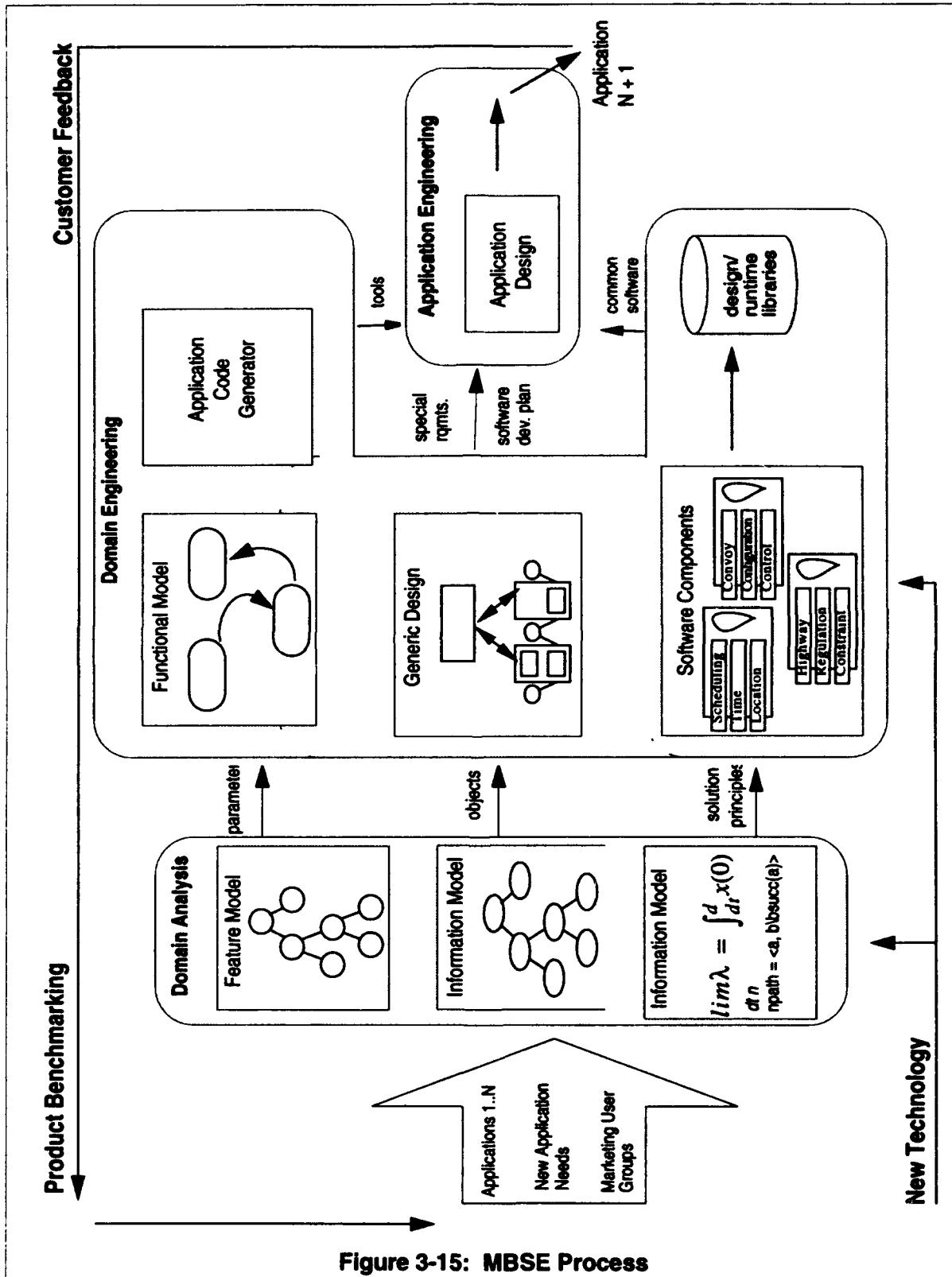


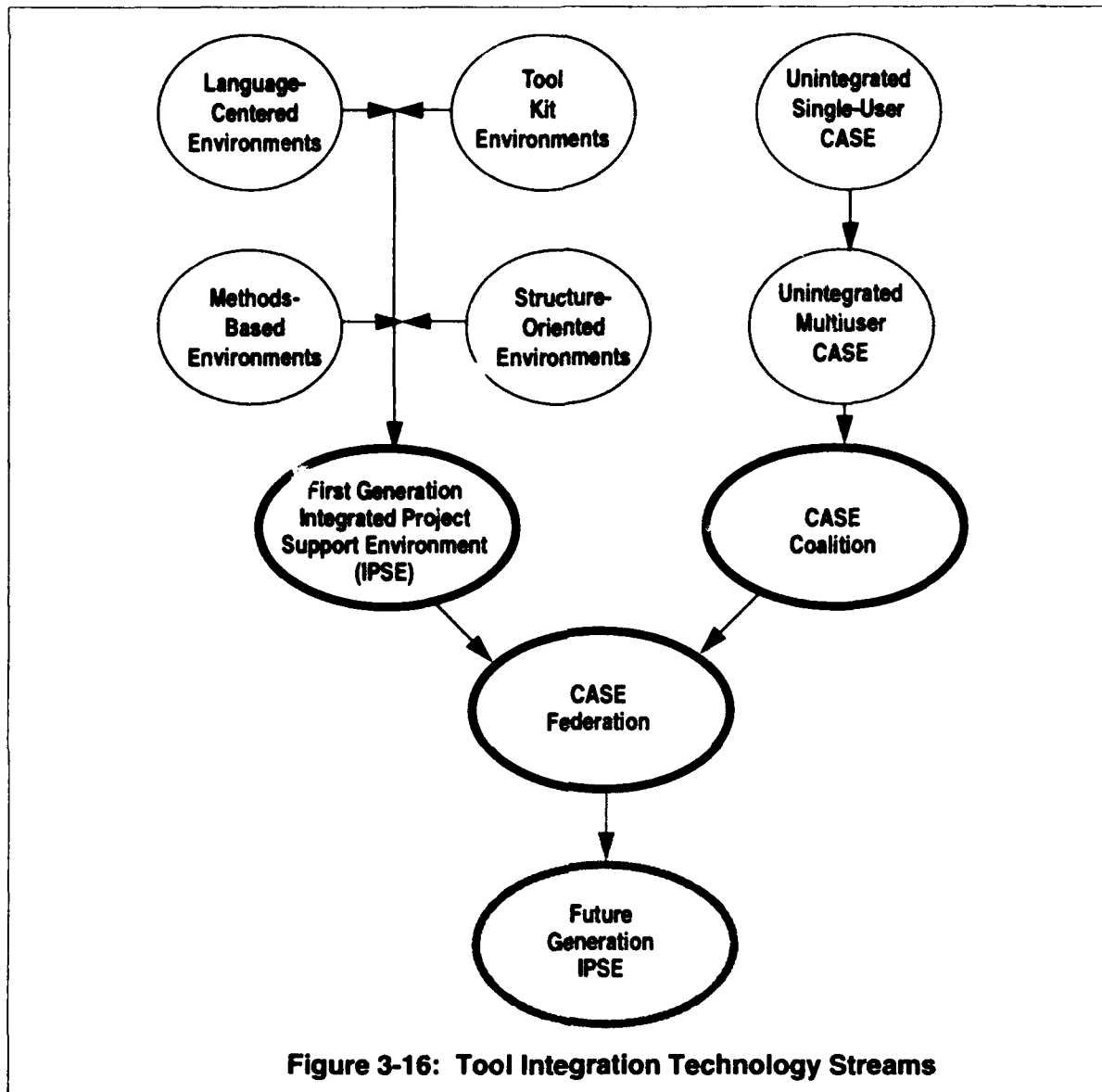
Figure 3-15: MBSE Process

Automation via Software Engineering Environments

In pursuit of the effective use of SEEs, the SEI is building on existing work in configuration management (CM) systems, environment architectures and integration, and CASE technology adoption. Together with the SEE community, we are evolving a conceptual framework for engineering SEEs that accommodates integration of rapidly evolving environment and tools technologies. Analysis of these technology trends will result in technology roadmaps. The result of this community work will be increased standardization in SEE interfaces and a systematic (engineering) approach to the provision of SEEs from commercial CASE components. Commercial system integration houses will establish business units offering SEE integration. Figure 3-16 shows a high-level roadmap of environment technologies representing both technology push and market pull. The roadmap illustrates that these two threads are on the verge of merging through a federated CASE architecture approach.

Our investigations into CASE in the context of the CMM and the emergence of computer-based process execution support will lead to the engineering of process-centered SEEs that are adaptable and flexible. In the long term, process-centered SEEs and intelligent engineering information access and learning technology may merge to provide intelligent SEEs for MBSE. The SEI has undertaken the following activities in support of this thrust area:

- Numerous TO&P arrangements that contribute to maintaining core competency, influence the SEE community (both CASE and SEE provider and user), and provide strategic advice. These include integrated CASE (I-CASE), SDIO, National Security Agency (NSA), STARS, DDI, and Next Generation Computer Resource (NGCR).
- Cooperation with industry partners such as International Business Machines (IBM) and Hughes, and collaborators such as Hewlett Packard, Paramax, and Digital Equipment Corporation.
- Assignment of resident affiliates to SEE work who both advance thrust area insights while they are at the SEI; and champion SEI strategic advice and act as transition agents back at their home organizations.
- Technical leadership positions in a number of forums including the North American Portable Common Tool Environment (PCTE) Initiative (NAPI), NGCR Project Support Environments Standards Working Group (PSESWG), NIST integrated Software Engineering Environment (ISEE), and the IEEE Technical Committee on CASE Adoption (P1348).
- Tracking related efforts, such as the American National Standards Institute (ANSI) Technical Committee (TC) X3H6 and X3H4, and CASE Communique.

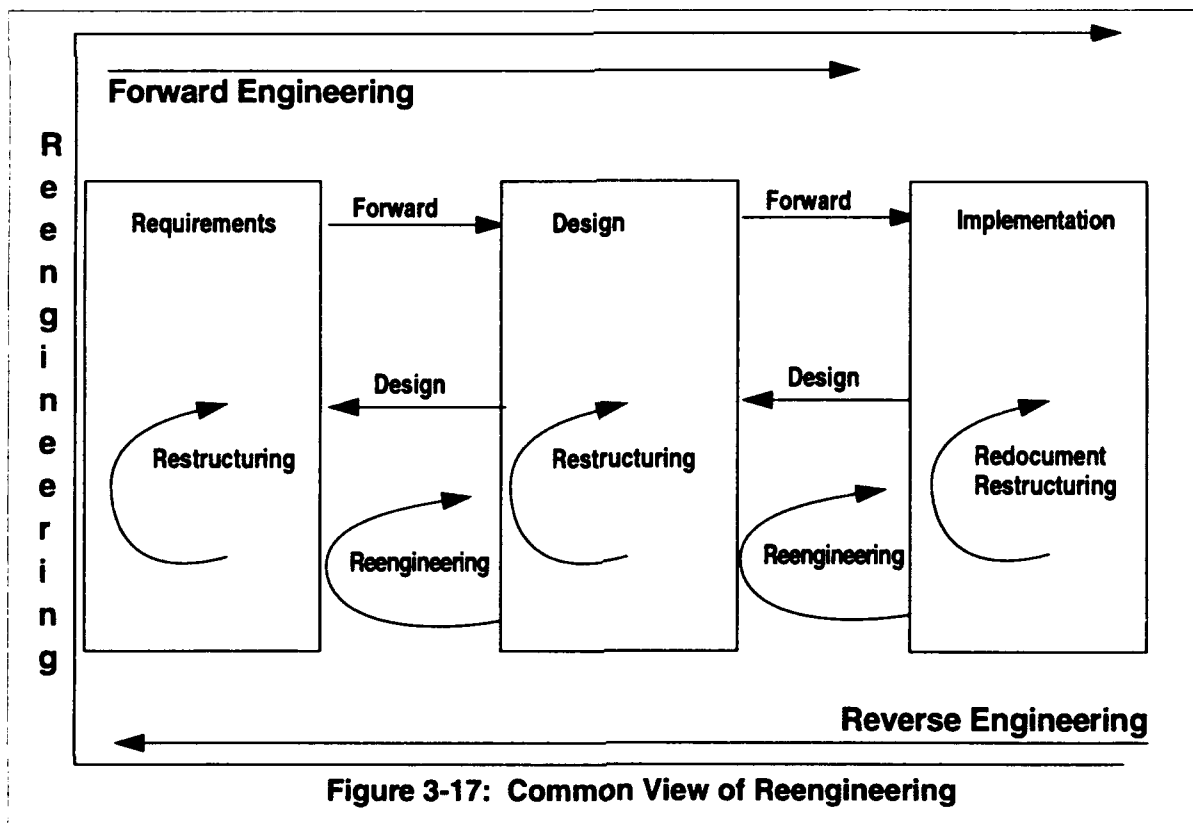


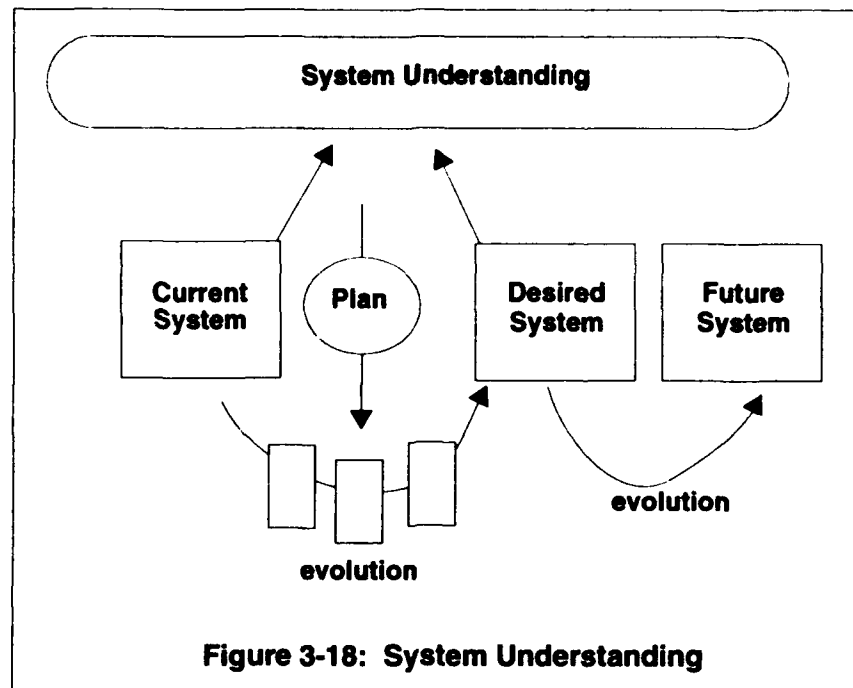
Reengineering

The initial focus on MBSE has been in application engineering through domain and architectural models with a reuse perspective, capture of engineering information in the form of requirements elicitation, and CASE adoption and assessment of environment integration technologies. Insights gained from the reuse focus have had an impact on Army reuse policy making. The original SEI work in Advanced Learning Technologies (ALT) and exploratory use of emerging commercial multimedia technology in software engineering provide the foundation for multimedia-based engineering knowledge codification. The thrust in SEEs has and will

continue to support the I-CASE acquisition by providing strategic insights and advice regarding best practice in SEE technology and in CASE adoption. Open systems issues play a strong role in advancing the state of CASE integration, where the SEI has leadership roles in standardization forums such as NIST, ISEE, and NGCR PSESWG.

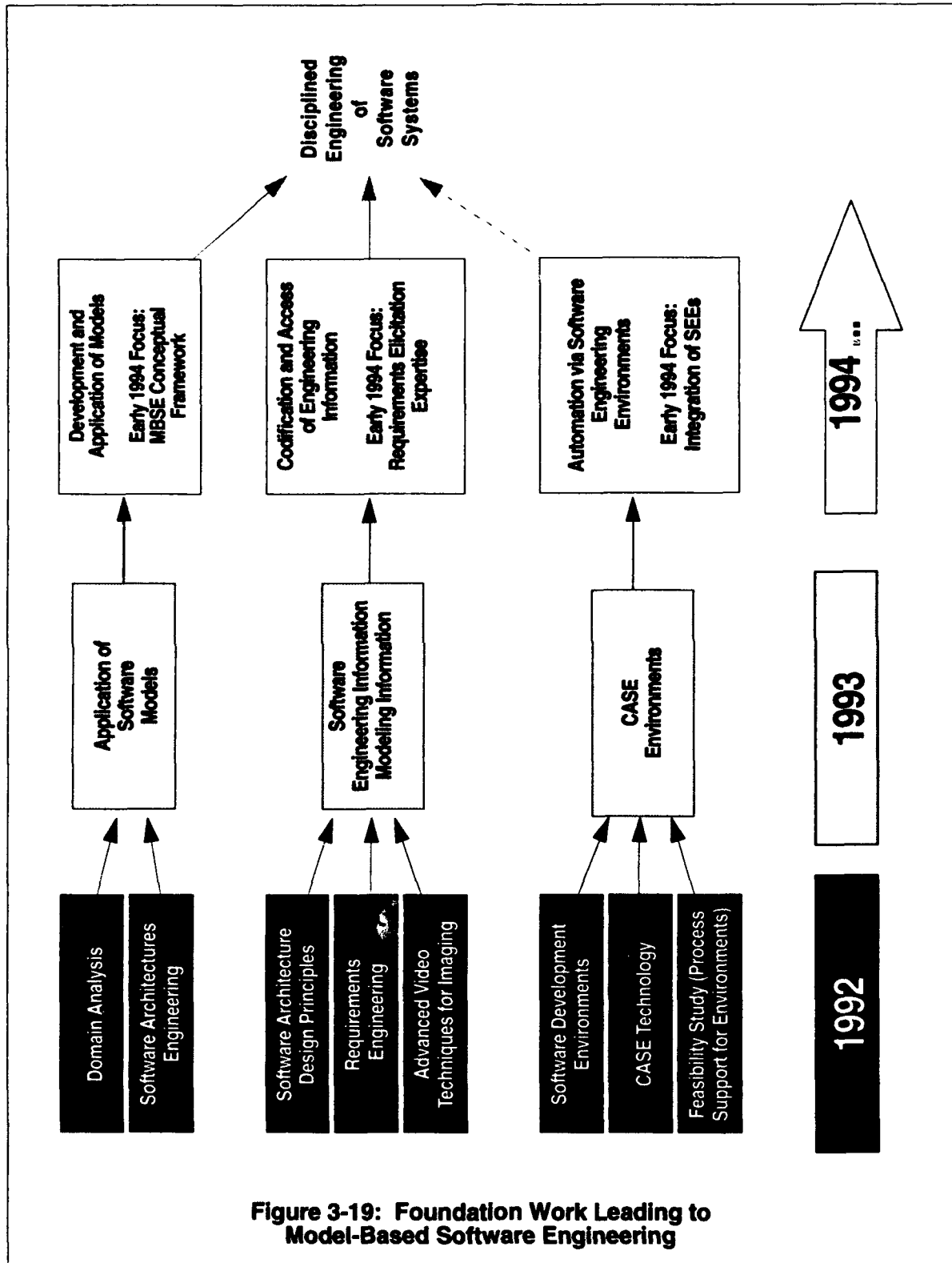
The SEI will be demonstrating responsiveness to changing external views by responding to customer interests, while maintaining a focus on an MBSE technology solution strategy. We will respond to customer interest by increasing emphasis on reengineering. A common view of reengineering is shown in Figure 3-17. Reengineering is viewed as a software engineering problem that can be addressed through a systematic problem solving approach. Figure 3-18 illustrates this view of reengineering. We expect to address it with contributions from MBSE in the form of the three thrusts in this focus area as well as technology results from real-time distributed systems (rate monotonic analysis (RMA), reliability and performance models), risk (risk analysis of reengineering alternatives); and process (reengineering process, reengineering economics supported by process and product measures). This involves cooperation with defense and civilian entities including the Joint Logistics Commanders Joint Policy Coordinating Group on Computer Management addressing reengineering policy making, and STARS demonstration projects piloting the application of advanced technology such as domain-specific architectures in reengineering existing systems.





Summary

This strategic focus on model-based software engineering practice has evolved from prior SEI work in methods and tools, summarized in Figure 3-19. The following section will outline current and future potential products that are grounded in this prior work in the area. The roadmap for these products (Figure 3-20) can be viewed as an extension into the future of the foundation activities outlined in Figure 3-19.



3.4.1.3 Potential Products

Methods and tools efforts lead to improvements in best engineering and reengineering practice and contribute to the codification of software engineering in the form of MBSE. This is a long-term activity. Intermediate technology results take the form most appropriate for the state of their maturity and targeted audience at the time.

Development and Application of Models

Tailorable Model-Based Software Engineering Course Collection (2Q 1994). This course collection builds on a 1993 product. Its purpose is to have a course package that can be tailored on short notice to specific customer interest in MBSE-related issues such as reuse, domain analysis, domain-specific architectures, and application engineering. A second dimension of tailoring is for managers, practitioners, and educators.

Guide to Best Model-Based Software Engineering Practice: Edition 1 (1995). This guide discusses engineering activities based on the application of engineering models spanning the life cycle, including requirements engineering, reengineering, and open systems. The guide will provide managers and practitioners with advice on selecting and effectively applying methods and models for a range of application domains and system implementations.

Guide to Best Model-Based Software Engineering Practice: Edition 2 (1998). This document will draw together the three thrusts of work in this focus area. It will reflect advances in application engineering based on models since Edition 1, and complement it with insights on best practice in SEE adoption, and intelligent engineering knowledge codification and deployment. It will have the character of an MBSE handbook. This product will provide managers with strategic advice on the suitability and expected benefits of MBSE, practitioners with guidance on its application, and educators with a reference document reflecting the state of MBSE and its support through environments.

Codification and Access of Engineering Information

Advanced Multimedia Organizer for Requirements Elicitation (AMORE) (4Q 1994). Requirements elicitation expertise will have been captured in cooperation with TI. The information will be available in intelligent multimedia form as well as in traditional training media (e.g., video). The multimedia technology-based prototype will demonstrate the feasibility of providing intelligent access to engineering information and knowledge in an interactive manner on media such as compact disk read-only memory.

Guide to Best Practice in System Understanding: Edition 1 (1996). This guide discusses system understanding from the perspective of software architectures, and techniques for capturing system information, including information abstraction technologies and automated system design records. This guide will provide managers and practitioners with strategic advice on advances in this software engineering technology area.

Intelligent SEEs (1998). The large amount of highly interrelated information required and generated in the engineering of large software systems and the complexity of today's SEEs overwhelm both users and existing technology. Progress in multimedia technology and multiple visualization models will offer a potential solution to this difficult problem. The purpose of this product will be to raise the awareness of managers, practitioners, and future technology producers to the potential benefits of an intelligent (MBSE) handbook concept integrated with SEEs.

Automation via Software Engineering Environments

Guide to Best Practice in SEEs: Edition 1 (1Q 1994). This guide will address managers and practitioners who will need an understanding of the best practice to effectively integrate tools into SEEs and to use SEEs to support the engineering process. The guide will provide a high-level roadmap of the major issues, together with pointers to additional resources. It will establish realistic expectations for users of CASE tools, and provide practical strategies for acquiring and integrating CASE tools and SEEs. It will consist of three components:

1. A strategy for the adoption of commercial CM systems; (in the form of a course, supplemented by a tape).
2. A framework and strategy to environment integration (course and environment integration book).
3. A strategy and approach to CASE adoption in relation to the CMM (document and tutorial).

Roadmap for Environment Technology (1995). Both managers and practitioners will require a conceptual framework for understanding and relating existing and future environment technology. The technology roadmap will provide a taxonomy of existing approaches, together with detailed descriptions of particular exemplars of each taxonomic category. Hence, the taxonomy will help fulfill an organization's near-term environment selection needs, and provide a framework for understanding future technology products.

Guide to Best Practice in SEEs: Edition 2 (1997). This guide will address managers and practitioners, who will need an understanding of the best practice to create and use effective process-centered SEEs, i.e., SEEs that understand and enact software processes. The guide will provide a high-level roadmap of the major issues, together with pointers to additional resources.

Reengineering

Annual SEI Software Engineering Techniques Workshop: Reengineering (3Q 1994).

This workshop is targeted to software engineering managers and practitioners. Each year it will focus on a particular topic that is of high interest to the customer community. The purpose of the workshop is to provide two-way communication between experts in the community and the SEI to identify a baseline of best practice on the selected subject. Previous workshops have been held on CASE adoption, CASE integration, requirements engineering, and reuse. The theme of the 1994 workshop will be reengineering.

Guide to Best Reengineering Practice: Edition 1 (1995). This document will provide advice on reengineering, including strategic approaches, state of the practice, decision support, risk assessment, method and tool adoption, and approaches based on domain models and domain-specific architectures.

Figure 3-20 shows the five-year technical product roadmap for the methods and tools focus area.

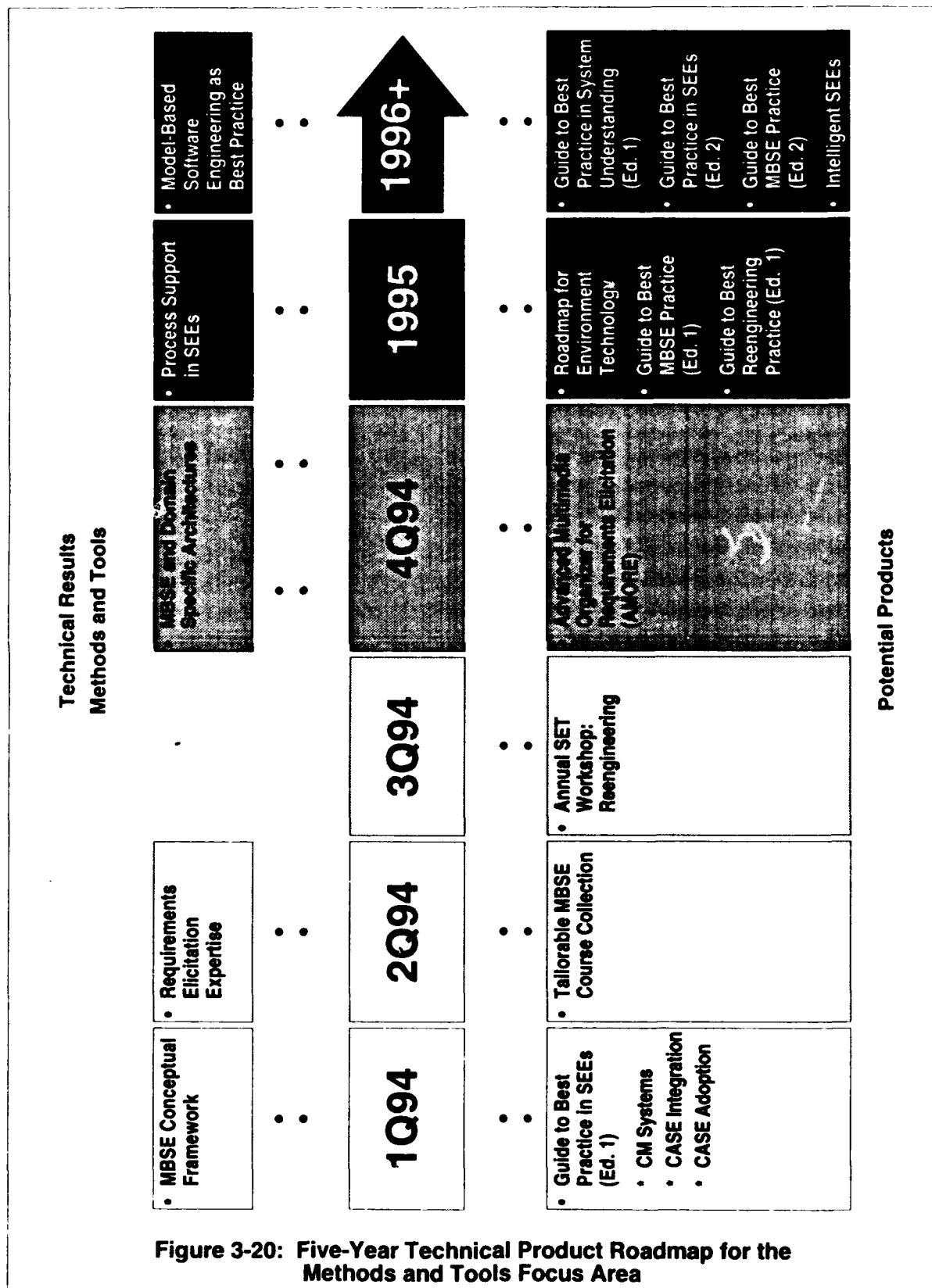


Figure 3-20: Five-Year Technical Product Roadmap for the Methods and Tools Focus Area

3.4.2 One-Year Plan

3.4.2.1 Context

During 1994 we will maintain focus on MBSE as outlined in the 1993 1&5 Plan. The current plan takes into account the execution of the 1993 1&5 Plan at 30 percent below the core funds indicated in the original plan. We will build on products and technical results from 1993. These include integration of domain analysis and structural modeling as a basis for demonstrating the feasibility of MBSE, insights into the state of CASE integration and adoption, and foundational work in multimedia engineering information modeling work as applied to requirements elicitation. The MBSE conceptual framework is the foundation for MBSE guides. 1994 will demonstrate our ability to maintain the stability of the technical strategy and direction, while being responsive to changing external views by responding to customer interest in reengineering.

To sustain critical mass and current staffing levels, we are complementing core with close to the same amount of TO&P funding. In 1993 we doubled the number of TO&P sponsors for that reason. We have been fortunate that in many cases the sponsors have accepted our technical strategy and are willing to contribute funds to our portfolio of technical work. The portfolio includes exploratory work, pilot projects, product development, and a limited amount of product delivery and consulting. This allows sponsors to benefit from the investments of their funds, and to complement those of other sponsors as well as the core funds. As a result, core deliverables are dependent on TO&P work, and the accomplishment of TO&P is dependent on core activities. In 1994 we expect to increase external sponsorship.

3.4.2.2 Core Funded Activities

Development and Application of Models

Core funds go toward a tailorable MBSE course collection, and the evolution and refinement of the MBSE conceptual framework, which is an essential component of the guide to best MBSE practice in 1995. The work on the MBSE conceptual framework includes an elaboration and documentation of an engineering process based on MBSE, investigations into domain and architectural taxonomies, and into characterization of desirable and undesirable system properties. Research efforts in this thrust are represented by the investigations into taxonomies and system properties. These investigations are also essential to a long-term involvement in reengineering strategies.

Codification and Access of Engineering Information

The initial focus is on demonstration of multimedia-based intelligent support for requirements elicitation. Core funds will be invested in the synthesis and application of multimedia technologies to demonstrate the practicality of multimedia-based intelligent software engineering support. We are cooperating with several other groups at CMU (School of Computer Science (SCS), Information Technology Center (ITC), and the EDRC) as well as a multimedia consortium between CMU, public broadcasting member WQED Communications Inc. in Pittsburgh, and Intel. This technology work is essential for the collaboration with TI on the AMORE. This technology is also the basis for a joint investigation with the real-time systems focus area on an interactive developer's handbook. These efforts incrementally build toward a large, multimedia software engineering knowledge base, leveraging the work on the MBSE conceptual framework.

The same underlying technology is of importance in the manufacturing arena in addressing its legacy engineering document problem. Through a combination of seed core and TO&P funding we are investigating the practicality of this technology in this context.

We are also investing in technical leadership roles in public forums such as international workshops and conferences sponsored by IEEE as well as the IEEE TC on multimedia technology. This allows us to leverage existing infrastructures to track and influence the advancement of the state of best practice in this technology area.

Software Engineering Environments

The focus for 1994 is on capturing and advancing the state of an engineering approach to CASE integration, and on assessing the next emerging commercial technologies in the form of computer-based process execution support.

Our assessment of best practice in SEEs is based on a conceptual framework for this technology area and consists of three parts addressing CASE integration, CASE adoption, and adoption of CM systems. In particular, we are investing in the publication of a book on the state of CASE integration to complement the potential product outlined in the 1993 plans.

Core funds are invested in advancing the state of environment integration. These include experiments in integrating different technologies (e.g., PCTE, A Tools Integration Standard (ATIS), Hewlett Packard Softbench, Sun Tooltalk, COBRA, ESF Kernel/2r) in a federated environment architecture, and technical leadership roles in public forums to foster technical consensus-building in SEE interface standardization. We are fortunate to have a complement of TO&P funds for these activities (see below). Core funded activities include participation in and

tracking of forums such as CASE Communique, ANSI TC X3H6 and X3H4. IEEE has chartered a TC on CASE adoption (P1348) with intentions to feed the results into ISO. Our CASE adoption work is the starting point for this TC, and we continue to provide technical leadership as technical editor.

The research component in this thrust focuses on the development of a conceptual framework for assessing emerging commercial technologies in support of computer-based process execution. Such technologies have their roots in process-centered environments (first generation integrated project support environments (IPSEs), STARS, Arcadia, ESPRIT, Eureka Software Factory (ESF), International Software Process Workshop series), in cooperative work and concurrent engineering support, in workflow and office automation, and in intelligent tutoring systems. This framework will be the foundation for technology and product development work in the out-years.

Reengineering

In response to ARPA guidance, the SEI proposes to provide a leadership role in defining and advancing best reengineering practice, taking advantage of current SEI activities in software engineering technologies, and related ARPA activities such as DSSA, STARS, the Virginia Center of Excellence, as well as Air Force activities such as CARDS, the Software Technology Support Center (STSC), and the Joint Logistics Commanders Joint Policy Coordinating Group on Computer Management addressing reengineering policy making.

The SEI proposes the following key activities to support the DoD reengineering of large software-intensive legacy systems:

- Develop a conceptual framework for reengineering. The conceptual framework supports identification and assessment of the maturity of software engineering technologies in support of reengineering. The technology results of this activity lead to products such as a guide to best reengineering practice, a reengineering technology roadmap, and a reengineering improvement strategy that leverages reuse and domain-specific architectures initiatives.
- Conduct an assessment of the state of the practice in reengineering. This product development activity involves community participation. It will benefit from the Annual SEI Software Engineering Techniques Workshop. The results will be incorporated into the guide to best reengineering practice and will serve as a baseline for improving reengineering practice.

- Accelerate the evolution of a taxonomy of domains and architectures and its reduction into reengineering practice. This research activity is essential to MBSE. Such a taxonomy is considered essential for successful evolution of a software technology base, and for effective identification and promotion of dual use software technology. Domain models and domain-specific architectures allow for more effective reengineering beyond code level program transformation.
- Identify and analyze desirable and undesirable system properties, their root causes, and their effects. These include properties of legacy systems as well as future systems. A catalog of such system properties is the basis for a systematic (engineering) approach to effective system understanding of legacy systems and their evolution strategies. An understanding of these properties is an essential ingredient of a systematic analysis of legacy systems for reengineering purposes. This activity has a product development component (codify known properties) and a research component (taxonomy of properties).
- Accelerate the evolution of the design record concept toward practical use. This research activity complements the work already outlined in multimedia engineering knowledge organization and is a critical element of maintaining system understanding. The design record concept provides a focus for capture, representation, and visualization of system information and knowledge. The SEI can become a critical link between innovative technology research as directly sponsored by ARPA, and its reduction into state-of-the-practice technology. The results of this task directly contribute to the evolution of a codified body of knowledge in software engineering, an essential element of a discipline.

The latter three items are research activities that continue beyond 1994. The proposed activities will not only accelerate advancement of reengineering practice, but also contribute to the improvement of software engineering based on models and architectures.

3.4.2.3 TO&P Funded Activities

Development and Application of Models

The methods and tools focus is continuing several external working relationships in which MBSE technology is being deployed. Activities include work in movement control with the Army Software Development Center, the tri-service construction of a radar to operationally simulate signals believed to originate within the Soviet Union (CROSSBOW-S) organization on the Joint Modeling and Simulation Systems (J-MASS), the BMD effort of SDIO, NIST work in reuse, and the Navy training effort. In these activities we provide a combination of technical leadership in MBSE based on domain models and architectures, training and education, and

strategic advice as input to reuse-oriented policies. Though having primarily focused on the reuse perspective of MBSE, several of these activities also have a reengineering character. Thus, these and other reengineering-specific TO&P activities will allow a quick start in reengineering issues.

Codification and Access of Engineering Information

In cooperation with TI we are capturing requirements elicitation expertise. This expertise will be made available in traditional training media (video). It is also the content material to demonstrate the practicality of intelligent access to engineering information and knowledge through multimedia-based technology.

The same underlying technology is of importance in the manufacturing arena in addressing its legacy engineering document problem. Through a combination of core and TO&P funding (NAVSUP, National Center for Excellence in Metalworking Technology), we are investigating the state of commercial information abstraction technologies from images for practical application in this context. Investment of core funds into the system design record concept has further potential for application in the manufacturing context.

Automation via Software Engineering Environments

We have TO&P sponsor interest in the publication of a CASE yearbook. We are investigating the possibility of STSC becoming the continuing publisher of such a yearbook.

Our continuing hands-on experimentation and analysis of CASE integration technology has strong TO&P sponsor support (including NSA, SDIO, I-CASE, DDI, STSC, STARS) as well as industry interest (IBM, Hughes). The insights gained provide the basis for CASE technology strategies. In particular, we are investigating the integration of multiple technologies into a federated environment architecture and the maturity of SEE interfaces for standardization in practice. Environment technologies include PCTE, ATIS, Hewlett Packard Softbench, Sun Tooltalk, COBRA, ESF Kernel/2r, and various advanced CASE products.

We have been involved in the I-CASE acquisition since 1991. We have performed a lessons learned study of previous government efforts in environments and held a workshop with I-CASE people to discuss successes and pitfalls. We have participated in the review of the request for proposal draft and provided training to the proposal evaluation team. We continue to provide strategic advice and insights both into environment technology trends and advances, and into CASE adoption—a critical element for the successful completion of the I-CASE effort. Regarding CASE adoption, the STARS demonstration projects offer an excellent case study in adoption of advanced SEEs into practice.

We continue to invest in technical leadership roles in public forums to maintain momentum in community consensus-building regarding standardization in relevant SEE interface areas. We have support from several TO&P sponsors for this activity. We are instrumental in resurrecting the NIST ISEE effort, which has produced the NIST European Computer Manufacturer's Association reference model for environment frameworks. We are developing partnerships with several government and industrial parties to maintain the transition infrastructure that exists as part of the NGCR PSESWG effort, but is in jeopardy for funding reasons. Both the NIST and PSESWG forums have attracted strong technical players from government and industry and are an ideal arena in which to provide technical leadership. We are providing technical leadership in the North American PCTE Initiative under sponsorship of DDI. In addition to these activities advancing the state of best SEE practice, we provide strategic advice, training, and consulting to our sponsors.

Reengineering

It is anticipated that we will have TO&P sponsorship for some of the reengineering-specific activities. These will complement the ongoing working relationships in model-based software engineering outlined above.

3.5 Real-Time Distributed Systems

Through its focus on real-time distributed systems, the SEI maintains its core competency in methods and tools for disciplined engineering of software for defense and civilian applications.

The civilian applications of interest to this focus area directly affect the nation's competitiveness and economy in addition to supporting defense needs. Because of their importance, during 1993 we broadened our attention from solely defense to include dual-use applications of real-time distributed systems technology. Examples of civilian applications include:

- Agile manufacturing facilities that are able to reconfigure plant operations quickly to meet changing requirements and to permit online maintenance and upgrade.
- Radars and other sensors for monitoring the development of weather patterns, seismic data, the status of power distribution grids, and the distribution of pollutants.
- Satellites, fiber-optic networks, and high-speed switches to transmit large volumes of live audio, video, sensor, and text data.
- Mass transport vehicles, airplanes and ships, oil drilling and refineries, and power generation plants.
- Patient monitoring, heart-lung machines, CAT scanners, magnetic resonance imaging, and other medical equipment.

The civilian applications are not restricted to public or commercial applications. Examples of government "civilian" applications include:

- Federal Aviation Administration (FAA) Advanced Automation System (AAS)
- National Aeronautics and Space Administration (NASA) Space Station Freedom (SSF)
- NOAA satellite networks

Numerous embedded systems for defense (surveillance systems; weapon systems; training simulators; command, control, communications, and intelligence systems, etc.) have similar requirements:

- The applications have long life cycles (decades rather than years) and require evolutionary upgrades.
- The applications assign paramount importance to quality attributes such as timeliness, reliability, safety, interoperability, etc.
- The applications require continuous or nearly non-stop operation.
- The applications require interaction with hardware devices.

Although business applications like a nationwide network of automatic teller machines evolve over years and provide secure services 24 hours a day, these requirements are soft: a failure (i.e., departure from specified behavior) does not have the catastrophic consequences of a chemical spill or a reactor meltdown.

Dual-use software technologies are critical components of our nation's infrastructure. Defense and civilian applications require continuous and widespread evolution: requirements change because the mission or the market changes; components (software and hardware) must be replaced when requirements change; systems are reengineered when components or requirements change; and finally, components and systems change when new technology is adopted to increase productivity and remain competitive. In other words, in these applications reengineering is the rule, not the exception.

The current state of the practice is unsatisfactory because systems and components use mostly proprietary technology and because designers lack quantitative methods for design. Use of non-standard, proprietary technology makes reengineering time-consuming and expensive. In particular, the system integration phase often requires shutting down the plant and deploying the equivalent of SWAT teams to conduct massive debugging sessions). The end result is that users upgrade (i.e., reengineer) their systems infrequently and lose efficiency, quality, and competitiveness.

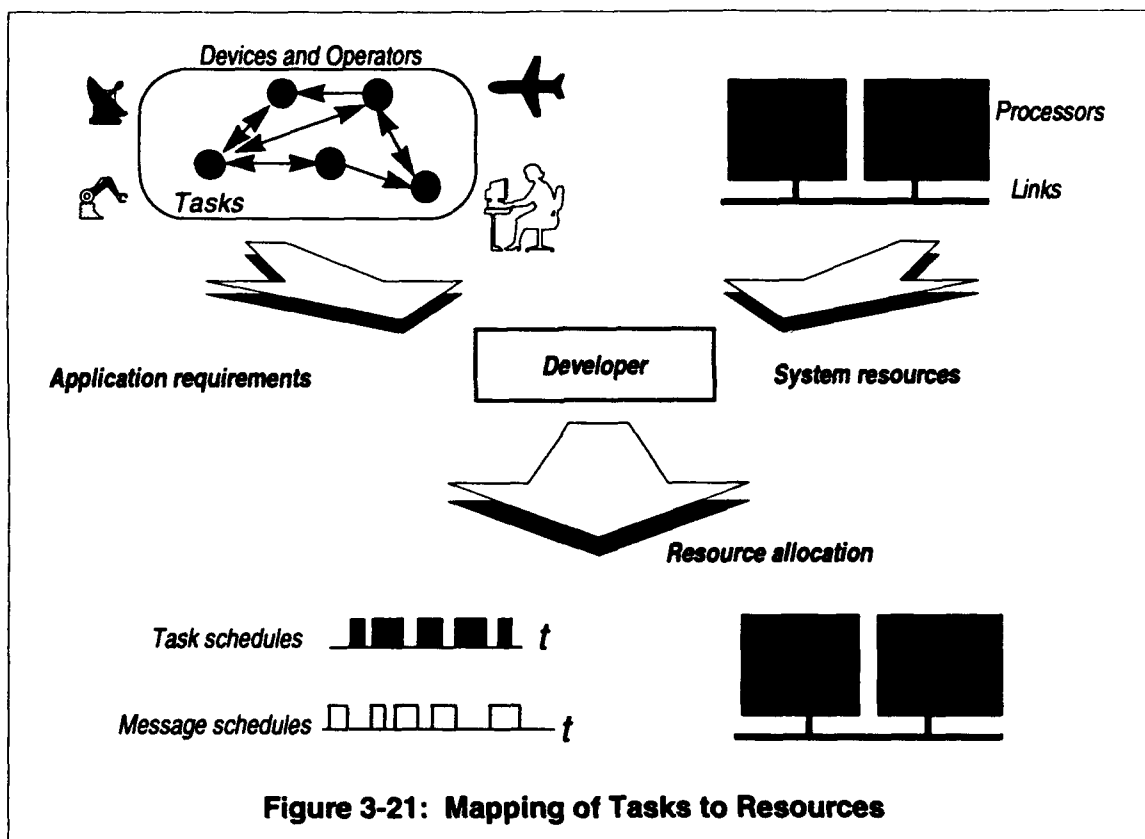
Anecdotal evidence gathered from interviews with developers and users of process control systems suggests a number of things that end users are unable to do today. They want to:

- Concentrate on selective software customization (using proprietary knowledge) of standard components built and integrated by others.
- Shop around for components and integrators to get better prices and services; users wish to achieve the state of the management information systems (MIS) community and their access to interoperable packages, servers, office equipment, etc.
- Upgrade by "roll-in/roll-out" of components to avoid down time, which is expensive or not available; the state of the MIS community is again seen as the desired state with its "plug-compatible" databases, word processors, peripherals, etc.

Achieving this state is important both in civilian and defense contexts. As indicated in Section 2.1.5, use of products and standards emanating from the civilian world offers an attractive way for the DoD to acquire and evolve systems of high quality at minimum cost and risk. DoD system acquisition procedures may change to allow more frequent use of commercial off-the-shelf (COTS) products. In using COTS, industry standards will become even more important to the DoD. Hence, the SEI must focus on the capability to design systems and their architectures using these standards.

Achieving the desired state will not be easy because of the built-in legacy of proprietary systems and components. Industry does not have ready access to COTS components that are interoperable and interchangeable while ensuring quality attributes such as timeliness, reliability, and safety.

Indeed, building interoperable and interchangeable components is not enough. Industry lacks widely accepted methods (e.g., integration guidebooks, models, theories) to quantitatively analyze, predict, or ensure system quality attributes prior to integration. Developers must understand how the various components of the system interact and what resources are needed to implement alternative designs. Figure 3-21 suggests this mapping of tasks onto computing and communication resources and the resulting task and message schedules.



To make informed design decisions, developers must quantify this understanding of components, interactions, resources, and alternative designs. Unfortunately, the software community lacks quantitative methods for the design of these demanding applications, and designers do not have tools to evaluate designs, upgrades, and tradeoffs. Cost and schedule overruns are common because serious problems are often not discovered until the system integration phase.¹

3.5.1 Five-Year Plan

3.5.1.1 Goals

The SEI is in a unique position to address civilian as well as DoD application needs because of its mission and experience in the maturation and transition of dual-use technologies developed for defense needs that can apply to the civil sector: CMM, risk assessment, software architecture models, reuse, RMA, CASE, etc. Within the real-time distributed systems focus area in particular, the long-term goal is that developers of civilian and defense systems will routinely employ quantitative methods to evaluate software engineering designs, upgrades, and tradeoffs. Moreover, the software community will view the SEI as a source of advice and expertise on the use of these methods.

Developers will use quantitative methods to design and construct software. These methods could be based on a theory or on empirical results. That is, developers could rely on tabulated, empirical results for which there is no well-understood theory. For example, before the theory of thermodynamics was developed, builders of steam engines learned from experience that some designs and materials could operate at certain limits of pressure, temperature, and humidity. Keeping within the limits produced reliable engines. When the limits were exceeded, engines exploded.

The handbook is one of the transition mechanisms used by the SEI (Section 4.1), and by 1999 the collection of accepted methods for evaluating and designing software systems will be codified in a collection of handbooks oriented to the needs of practitioners. The format or style of the handbooks will evolve over time, as users gain experience with them and we adopt lessons learned from our investigations into technology transition (Section 4.1.1).

¹ Horror stories are not uncommon. "On Monday, September 2, 1991, at 9 a.m., a \$6 million manufacturing support systems/network integration program went live, the largest computer project the company had undertaken. By 10:30 a.m., 'The system had miserably failed,' reported the program manager of a chemical company. 'We had not anticipated needing so much memory, consequently, the system froze in less than two hours, stopping all work at the site.'" [*Enterprise*, Digital Equipment Corp. Vol. 6, No. 4, April 1993]. The quickness of the disaster suggests that the designers were flying blind, so to speak, throughout the development of the system.

The goals for 1999 are to:

- Identify, assess, and mature promising dual-use technologies to analyze, predict, or ensure quality attributes of a defense or civilian system.
- Demonstrate and define the best practices using these technologies.
- Develop handbooks for end users and integrators that identify best practices to analyze, predict, or ensure quality attributes and offer roadmaps to adopt these best practices.
- Contribute to the development of open standards for suppliers that support the best practices and offer roadmaps to adopt these standards.

In pursuing these goals, we will develop a transition infrastructure that includes vendors, professional organizations, educators, and researchers to assure appropriate and ongoing use of these design concepts and quantitative methods. The 1999 goals are a refinement of the 1998 goals described in our previous plan. The 1999 goals reflect our expanded orientation to dual-use technology for defense and civilian applications and a sharper focus on the quality attributes of the systems.

The success of the SEI in the real-time distributed systems focus area can be measured by the application of technology matured by the SEI within the systems produced for dual-use systems. By 1999, there should be official standards in place in open systems and in high-speed switching networks that reflect matured real-time technology. By 1999, all flight simulators should include structural modeling technology, and science and technology maturity models should have achieved the same level of popularity as the CMM has currently. Also by 1999, the impact on process control should be seen in the number of civilian and defense software developing organizations that use SEI-matured technology. The number of contributors to and users of SEI handbooks will provide a key indicator of success since the handbooks will act as the repository of the state of the practice. The contribution by others to portions of the handbook will indicate the dual-use of that repository.

In the short term, success measures are grades on TO&P reports that measure SEI effectiveness with its sponsors; the number of industry participants in joint work, which measures the beginnings of our impact on defense and civilian sectors; and the popularity of science and technology maturity models compared to the CMM at an equivalent stage of maturity.

3.5.1.2 Strategy

A strategic intent of the SEI is to help customers make lasting improvements to their overall software engineering capabilities. By focusing on defense and civilian applications, we concentrate on the problem introduced by the lack of quantitative methods of design. The strategy we will follow is to develop and deploy tools and techniques that address the following concerns:

- How to be more precise in stating performance, reliability, availability, safety, security and other quality requirements of software components and systems.
- How to model and predict component and system behavior before development or modification.
- How to configure and schedule hardware and software resources to meet system quality requirements.
- How to safely upgrade hardware and software components without shutting down the entire system.
- How to use open system standards effectively in defense and civilian applications.
- How to reengineer software for defense and civilian applications.

Current partners in this endeavor include multiple DoD organizations such as the NGCR; Office of Naval Research (ONR); Army Simulation, Training, and Instrumentation Command (STRICOM); Air Force Training Simulator Office; Defense Modeling and Simulation Office (DMSO); and the Ada Joint Program Office (AJPO). Additional partners include other government organizations such as NASA Space Station Freedom, NASA Goddard Software Engineering Laboratory, and the Nuclear Regulatory Commission (NRC); other federally funded research and development centers (FFRDCs) such as MITRE and Lincoln Laboratories; industry partners such as IBM, Digital Equipment, TI, Raytheon, and Hewlett-Packard.

3.5.1.3 Potential Products

The technology projects supporting work in this focus area contribute to the maturation of design concepts and quantitative methods and the codification of these concepts and methods in developer handbooks. This is a long-term activity. Intermediate results will take the form of courses, awareness lectures, technical reports and articles, software and proof-of-concept demonstrations. The primary objective of these intermediate "products" is to gather information about best and current practices, to affect these practices, to increase community awareness of these practices, and to test design concepts and quantitative methods of design before they are included in the handbooks.

The current mix of potential products of this focus area can be classified into four groups: open systems standards technology, dependable systems technology, structural modeling technology, and technology awareness. Standards efforts represent an important type of SEI service and a high-leverage activity for improving the state of the practice since they are community efforts, developed and distributed by organizations such as IEEE, ANSI, and ISO.

Open Systems Standards Technology

Open Systems Standard for Real-Time Communications (POSIX 1003.21) (2Q 1994, 1995). With funding of the Navy NGCR, the SEI participates in, and in some cases (POSIX 1003.21) leads, the development of open, dual-use standards that meet the requirements of the mission-critical community. POSIX 1003.21 will be a standard suitable for the real-time distributed systems communication domain. Our efforts ensure that client needs are met while contributing to the larger aims of the focus area. During 1994 drafts of POSIX 1003.21 will appear, and the final standard is expected to be approved in 1995. The development of standards is a time-consuming process. They often take years to reach official approval with multiple intermediate drafts circulated for comments and voting by the technical community. This is also the target date of the Open Systems Architecture Handbook.

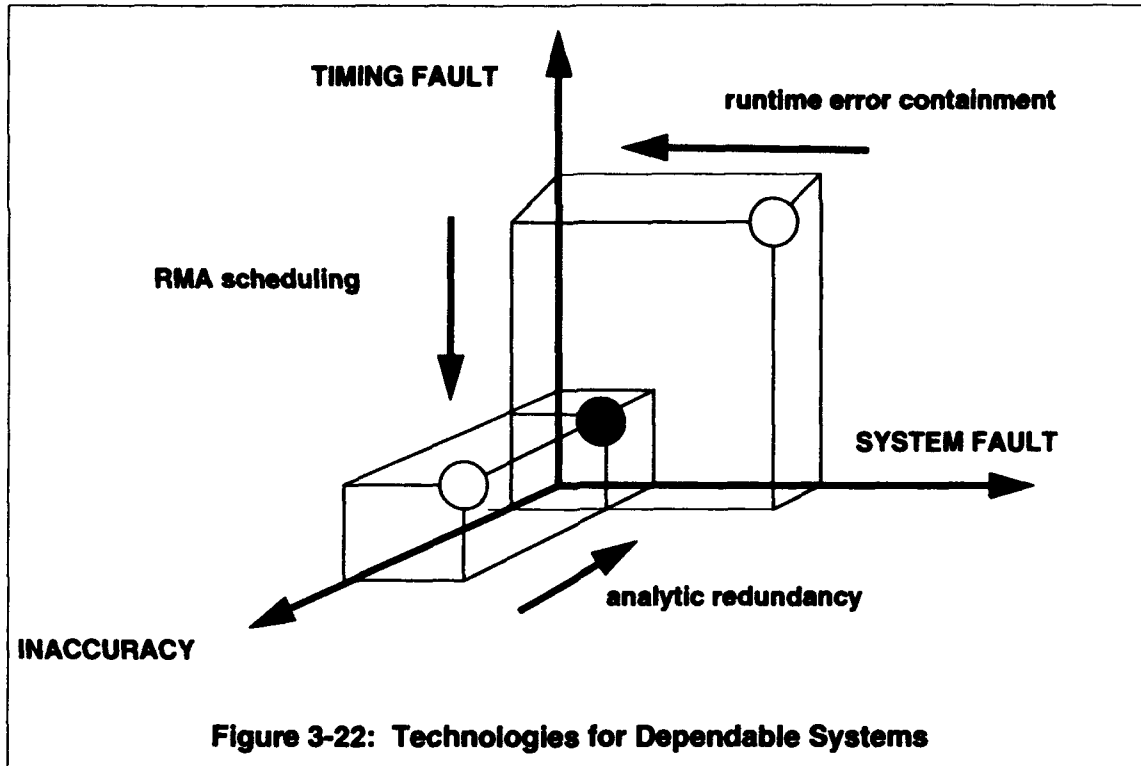
Open Systems Architecture Handbook (1995). With cooperation of NGCR we plan to develop a handbook for using open systems architectures in mission-critical systems. This handbook will be architecture-based and will highlight issues and approaches for solutions to the major problems facing developers.

Open Systems Standard for High Performance Networks (TBD). During 1993 NGCR and the ONR are planning to task the SEI to lead the establishment of a high performance network standard that can be used in both Navy and civilian applications. This new standard will be an extension to commercial standards and will complement the development of the software architecture guideline. No target date for this standard has been established.

Dependable Systems Technology

Systems Fault Tolerance Handbook (3Q 1994). Fault-tolerant systems incorporate features that recognize faulty behavior, contain the effects of faults, and recover from damage that may have been caused by the faults. We seek to develop, refine, and disseminate a conceptual framework for fault-tolerant computer systems. During 1994 we will develop a preliminary Systems Fault Tolerance Handbook.

Airborne Radar Study Reports (4Q 1994). We are conducting research on integrating the theory of analytical redundancy with the theory of generalized rate monotonic analysis and reducing them to a standardized quantitative method for real-time fault-tolerant computing (Figure 3-22). This technique, which we call the Reliable Application Kernel, is being used in a proof-of-concept demonstration in cooperation with another FFRDC (MITRE) working for the Airborne Warning and Control System (AWACS) Program. The results will be published as Airborne Radar Study Reports.



Structural Modeling Technology

Real-Time Simulators Guidebook (3Q 1994). With Air Force funding (ASC/YT), during 1994 we plan to develop the second edition of the Real-Time Simulators Guidebook.

STRICOM Reports (4Q 1994). The structural modeling technology will be used in other domains. With Army funding (STRICOM), planned future results in this area include reports on Close Combat Tactical Trainer (CCTT) Specification, WARSIM 2000 model extensions, DIS model repositories.

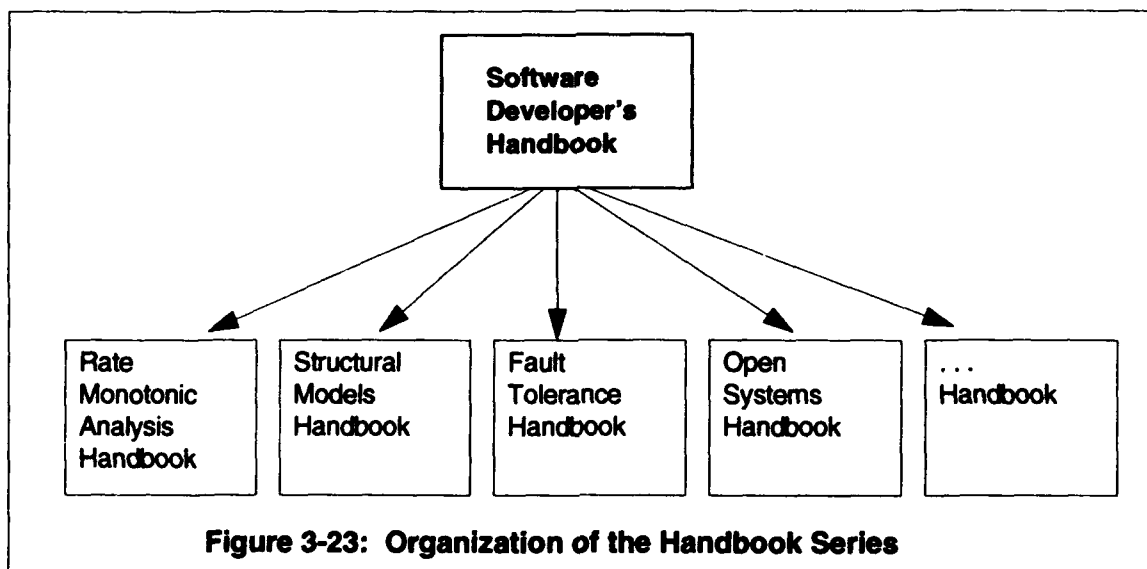
Core Architecture for Flight Simulators (1996). During 1993 the Air Force formed a systems engineering team composed of Air Force, SEI, and Link engineers to develop a core architecture based on structural modeling concepts. This architecture will serve as the basic structure around which most fixed wing flight simulators that ASC/YT develops will be based. The architecture will be extended with specific capabilities required for various aircraft (F16, F15, F22, etc.) to provide an extended core for the aircraft family. Customized extensions will be added for variants within the same family (United States Air Force, foreign military sales, etc.). This effort spans three years overall. After the core architecture is developed, the work will progress to developing F16 core extensions and then extensions peculiar to a specific F16 simulator.

Technology Awareness

Software Developer's Handbook (annual). During 1994 we will prototype the first edition of the Software Developer's Handbook. The handbook will document the best accepted practices in defense and civilian applications and collect these accepted practices in a single, convenient repository. The handbook will provide a source of ready information to practitioners that will assist them in their day-to-day work. The scope of the handbook will be limited to information required to perform relevant tasks, based on an analysis of practitioners' needs. As a reference book, it should explain how to perform the identified tasks; it will not contain detailed information concerning why tasks are performed in a certain manner. The handbook will describe:

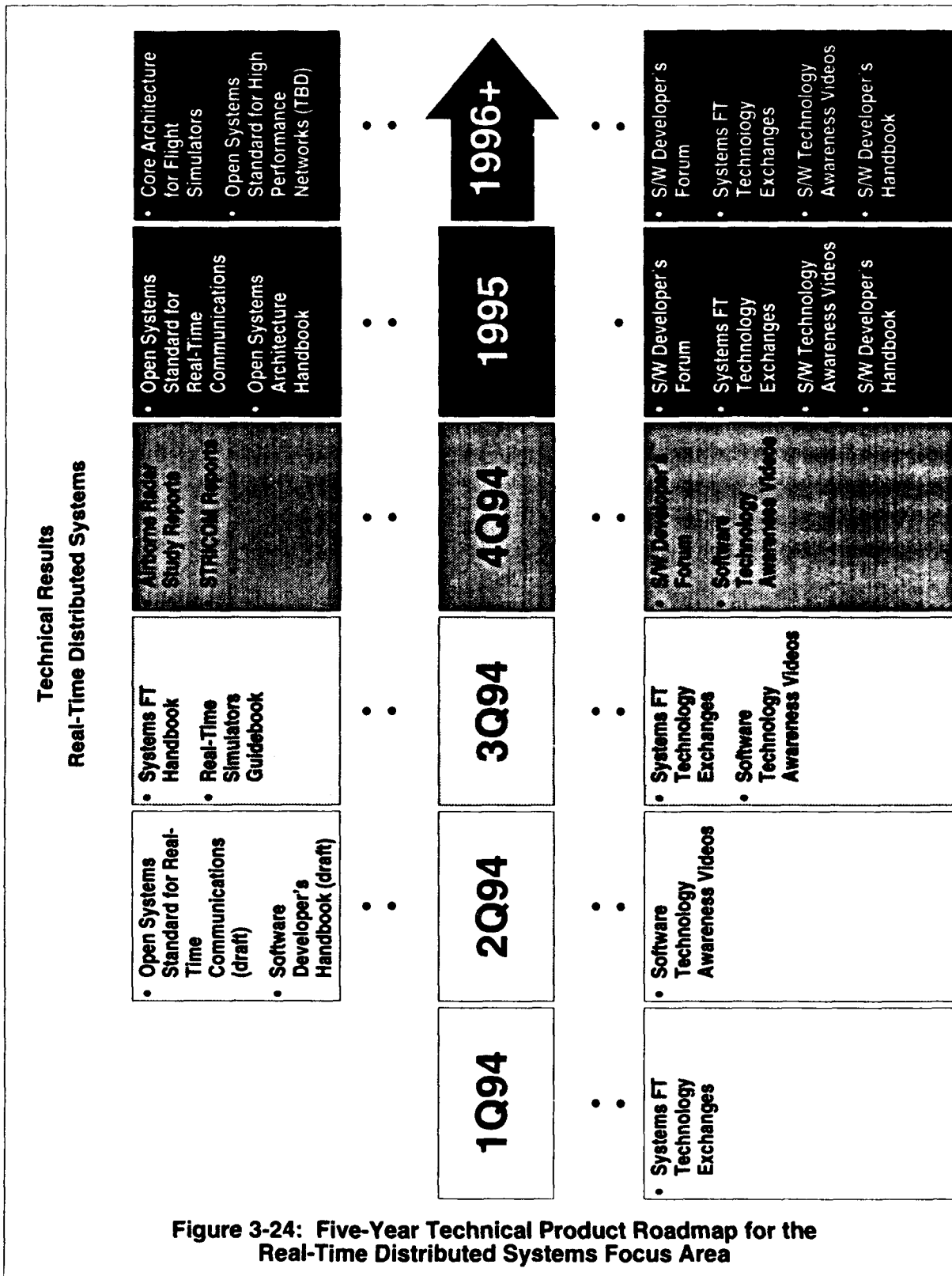
- What technology is available, i.e., the best practice.
- What the technology can do for the user, i.e., the maturity of the technology.
- When to use the technology.
- What is expected of the user, i.e., organizational processes, training, experience.

The handbook should be sufficient for a user to write a project plan: scheduling activities, selecting techniques and tools, adopting a software development process, etc. It will describe what technology is available for use. How to use the technology is the subject of separate technology-specific handbooks as illustrated in Figure 3-23.



To complement the handbook, additional products in this category include software technology awareness lectures (e.g., the Systems Fault Tolerance Technology Exchanges started in 1993), an annual *Software Developer's Forum*, and technology- or domain-specific SEI guides to best practices.

Figure 3-24 shows a five-year technical product roadmap for the focus area.



3.5.2 One-Year Plan

3.5.2.1 Context

Dependable Systems Technology Products. During 1993 we published a Fault Tolerance Framework Report, a standard vocabulary for describing system dependability and safety-critical requirements, faults, hazards, fault phenomena, and system-level fault tolerance approaches. This framework is a preliminary step to the Fault Tolerance Handbook planned for 1994 and beyond. In the software engineering arena, a system is often equated with software, or perhaps with the combination of computer hardware and software. Here, we use the term system in its broader sense. As shown in Figure 3-25, a system is the entire set of components, both computer related, and non-computer related, that provide a service to a user. For instance, an automobile is a system composed of many hundreds of components, some of which are likely to be computer subsystems running software. A system exists in an environment (e.g., a space probe in deep space), and has operators and users (possibly the same). The system provides feedback to the operator and services to the user. Operators are shown inside the system because operator procedures are usually a part of the system design, and many system functions, including fault recovery, may involve operator action. Not shown in the figure, but of equal importance, are the system's designers and maintainers.

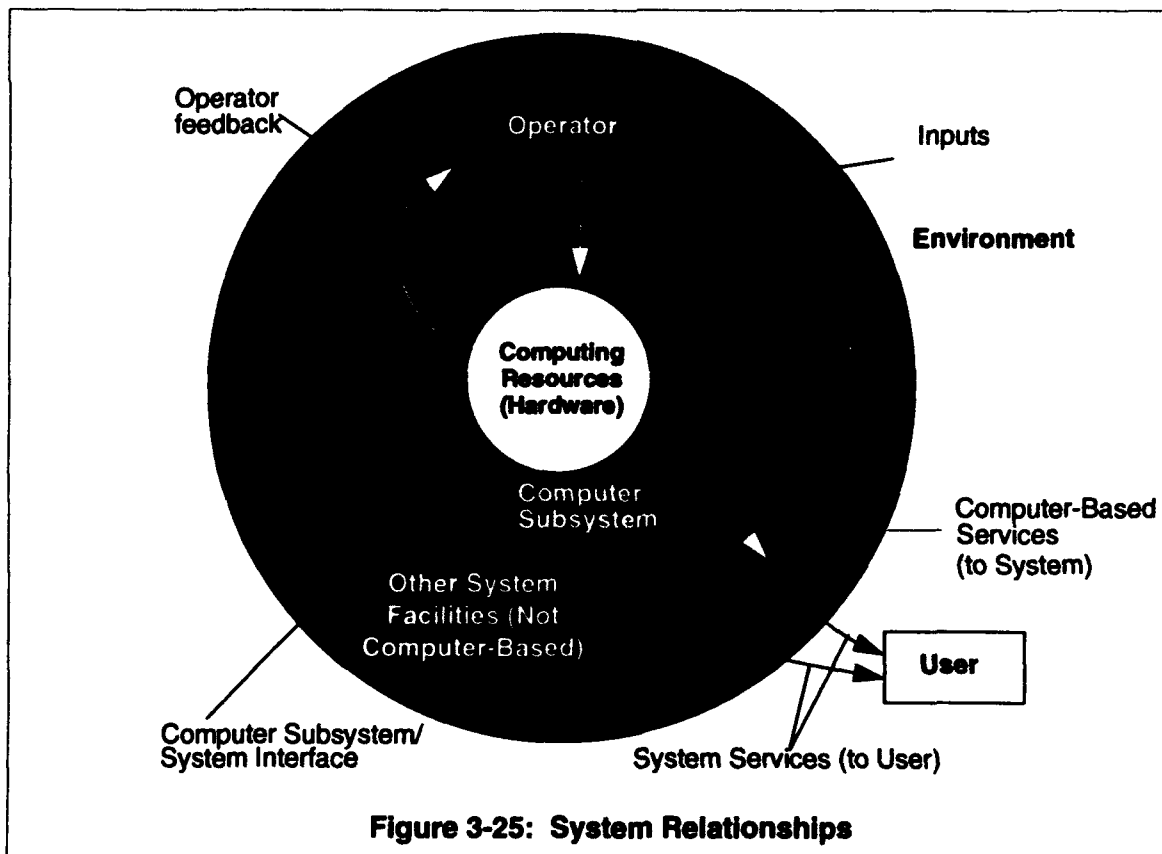


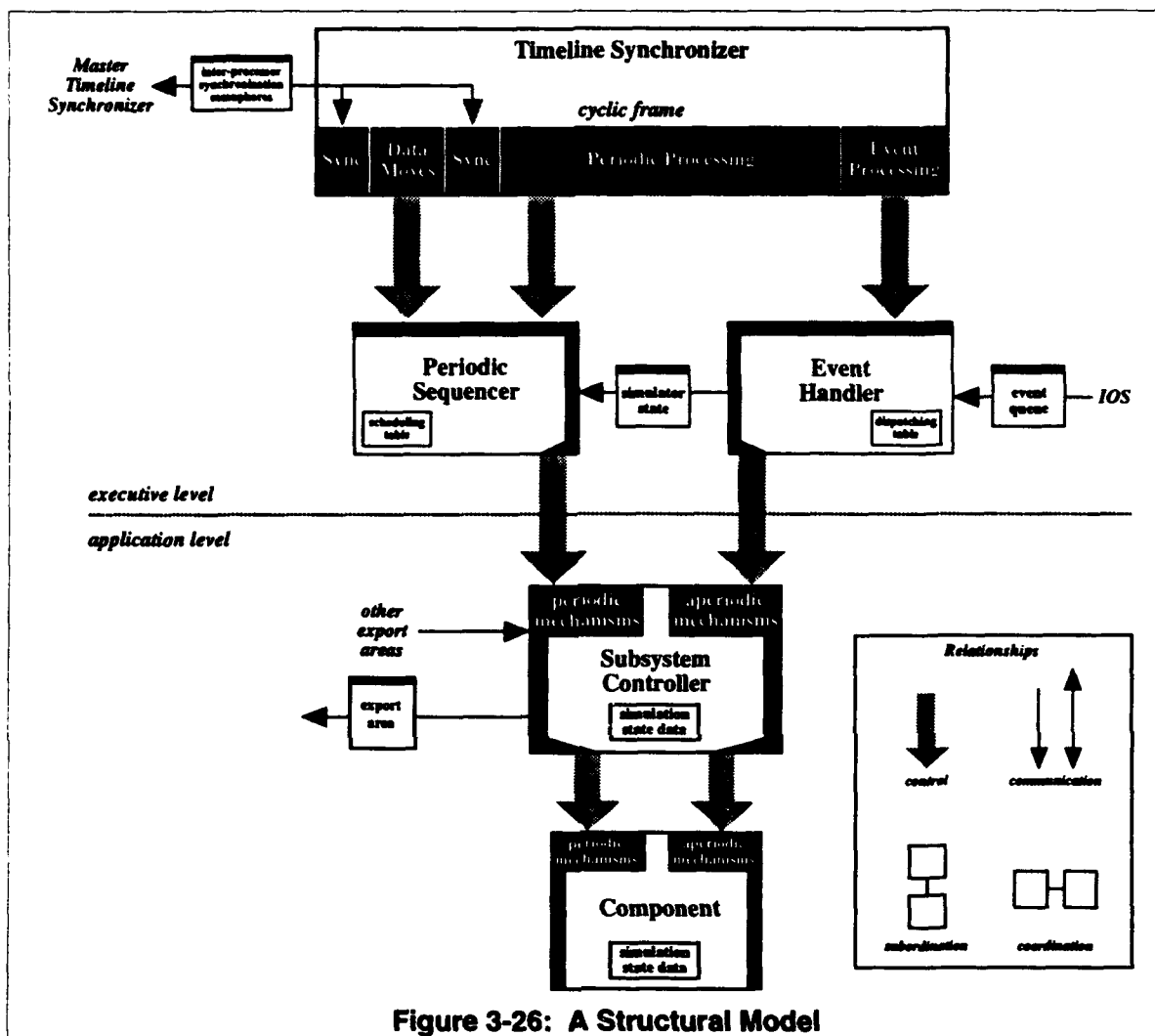
Figure 3-25: System Relationships

Structural Modeling Technology Products. During 1993, we published the first edition of the *Real-Time Simulators Guidebook*, a report that describes a validated structural model for real-time simulators including vehicle simulation, instructor/operator station, and environmental models. Figure 3-26 shows an example of a structural model. This figure demonstrates the structural model used in the air vehicle portion of flight simulators. The figure shows the various structural components and the data and control flow relationships among them. The executive portion is responsible for communication with the other portions of the simulator, the management of time, and the assignment of control to the application portion of the simulator. The application portion is composed of instances of subsystem controllers and components that communicate with each other in a well-defined, constrained fashion. A portion of the power of the structural model comes from the uniformity introduced by constructing a system as instances of a small number of structural types and by imposing discipline on the communication and control relationships. Structural modeling is evolving to incorporate emerging technology and other domains, as illustrated by our report on distributed interactive simulation (DIS) architectural modeling (1993).

Technology Awareness Products. During 1993 we completed the first edition of the *Rate Monotonic Analysis Handbook* and it was published by Kluwer Academic Publishers. The handbook has stimulated consultants and training organizations to include RMA within their repertoire of skills and courses. During 1994 we will continue the maturation of RMA by integrating it with analytic redundancy theory as a fault tolerance technique.

During 1993 as part of our research effort, we initiated a feasibility study on science and technology maturity models comparable to the CMM underpinning the process focus area. A science and technology model could be a guide for improving the scientific and technological foundations underlying software engineering. The focus of this model would be on stimulating the maturation of science and technologies and on finding scientific technological gaps that have direct impact on building various classes of software artifacts.

A science and technology maturity model would improve the knowledge base underlying our understanding of properties of software artifacts. The model would characterize the state of the art for specifying, analyzing, and optimizing properties of these artifacts. The analogy to CMM should not be misinterpreted because "levels" suggest an ordering or prioritizing for maturing technological know-how. Technology evolution does not always progress in an orderly fashion. Maturation is an iterative process rather than a linear sequence of "maturity levels" (Figure 3-27). During 1994 we will continue our research by starting to validate the models by characterizing the state of the practice in defense and civilian applications.



During 1994 we will continue developing quantitative methods and tools in three technology areas: open systems standards, dependable systems, and structural modeling. The potential products resulting from these activities were described in the previous section.

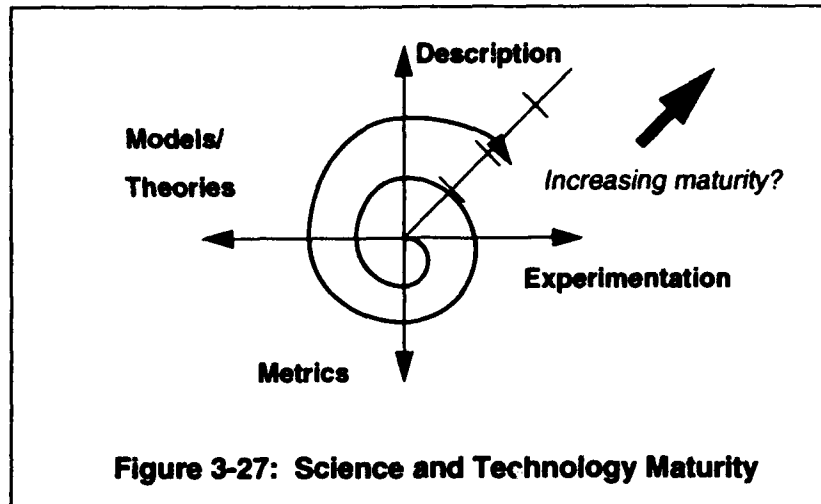


Figure 3-27: Science and Technology Maturity

3.5.2.2 Core Funded Activities

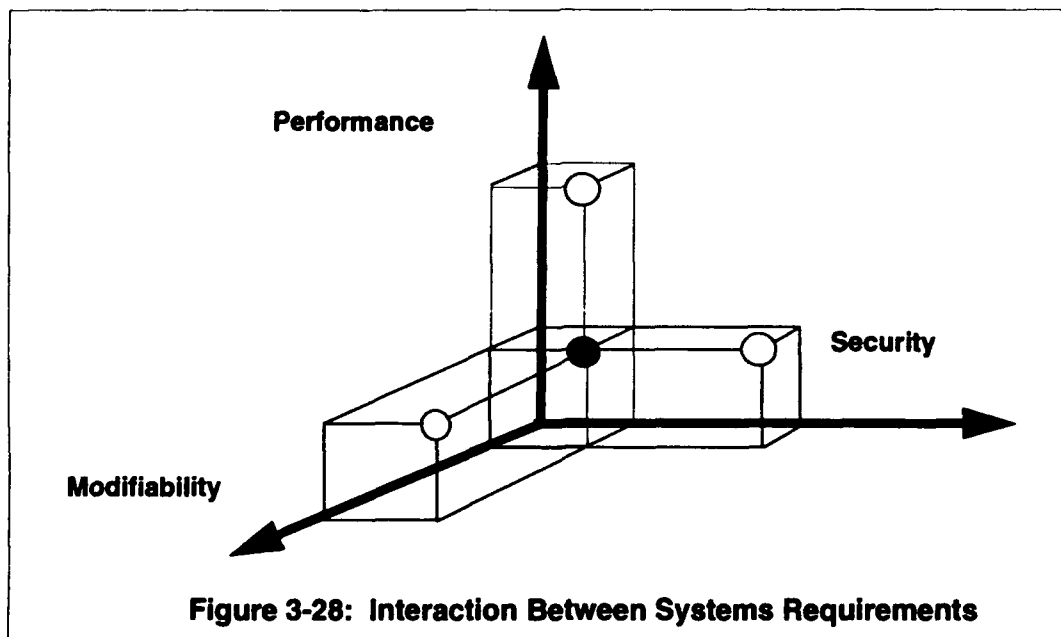
The bulk of core funding is dedicated to technology maturation and transition activities. The technology areas include open systems, fault-tolerance, structural models, and benchmarking that lead to potential products, as described earlier. In addition, as these potential products are completed and delivered to the user community through various SEI transition mechanisms, we identify and assess new targets for technology maturation through feasibility studies. During 1994 we will carry out several feasibility studies on emerging dual-use technologies for defense and civilian applications. The selection of feasibility studies will depend on the allocation of core funding.

Feasibility Study on Structural Modeling for Process Control Systems. The objective of the project is to demonstrate and transition a technology developed for a defense application into the civilian domain of process control. The state of the process control industry is similar to that of flight simulators when structural modeling was first developed. The engineers are predominantly domain experts with mechanical, chemical, or electrical engineering backgrounds, and few have extensive training in computer science and software engineering principles. The systems range from small to very large, with increasing scale and levels of integration, internally as well as with corporate information resources. The systems are long lived and subject to extensive modification. Existing systems are collections of disparate pieces with no systematic approach to architecture. The design is ad hoc, resulting in brittle systems that break easily under modification and are expensive to maintain and extend. Much effort is expended integrating components that have different interface requirements and expectations about the environment they run in. Based on our experience with the development of structural models for flight simulators, the architectural approaches we have developed could be applied effectively to this new area. The problems of real-time resource allocation and

management and scale are the same ones we have seen before. Process control is somewhat simpler from a computational viewpoint in that most of the applications within the process control domains are not as complicated as a flight simulator. However, precision positioning machinery is quite challenging, and the pattern recognition used in inspection systems is advanced. This activity maintains the SEI core competency in software technology transition.

Prototype Interactive Software Developer's Handbook. Technology advances will make the Software Developer's Handbook a live document, requiring continuous updating. This will require technology not only to update the information repository but also to organize it along a "virtual table of contents," customizable by the users. The prototype will be developed in conjunction with other SEI staff as an electronic multimedia handbook demonstration.

Feasibility Studies on Analysis of Quality Attributes. During 1994 we will conduct research on current and emerging technologies for integrated system development. A major problem with delivered systems is that they do not meet the user's needs because the designers have been too narrowly focused on meeting some goals without consideration of the impact on other goals. For example, a designer may focus on achieving particular performance objectives without consideration of the impact on modifiability. What is needed is a means of analyzing the trade offs between the various quality attributes so that designers can evaluate a design from the point of view of multiple user needs rather than a single one, as suggested by Figure 3-28. The ultimate goal is a methodology to quantitatively analyze, predict, or ensure product characteristics of the system. Such methodology will allow developers to evaluate and trade off the quality attributes to arrive at a better overall system.



3.5.2.3 TO&P Funded Activities

TO&P funded activities in this area are closely connected to core funded activities. We use core funding to mature and transition new technology, often in support of "delivery" to a TO&P sponsor. TO&P funded activities have already been described in previous sections:

- Air Force (ASC/YT), Army (STRICOM), and DoD (DMSO) fund structural modeling
- DoD (SDIO) and Navy (ONR) fund dependable systems
- Navy (NGCR and ONR) funds open systems standards

4 Technology Transition

The SEI mission of advancing the state of software engineering practice requires a technology transition strategy that enables us to “work smart,” one that gives us leverage in meeting the needs of our customers. We have learned that development and delivery of products and services offer the most effective way to accomplish transition. Our transition products and transition services help our customers make lasting improvements in their ability to acquire, develop, and maintain software-dependent systems and in their ability to educate people to perform these activities more effectively. We also act as a catalyst that stimulates the growth of an infrastructure owned and maintained by the software community. This approach to transition enables us to have the greatest impact with our limited resources.

In Section 4.1, we describe the models on which our transition strategy is based. These models enable us to approach transition in a systematic way and provide a variety of products and services that address our customers’ various needs. As we further investigate transition models (see Section 4.1.1), we will refine our strategy and expand our set of products to gain the greatest leverage possible.

In Section 4.2, we briefly describe the types of SEI products and services available from the SEI and the customers who use them: managers, practitioners, and educators. This section includes a list of products planned for 1994 and potential products to be developed by 1999. Delivering these products and services to the software engineering community is a challenge for a small organization like the SEI. To gain leverage, we work with transition partners, who assist us in tailoring and delivering our products. Section 4.3 describes the activities of transition partners in more detail.

In addition to developing relationships with transition partners, we have a range of other relationships that give our customers opportunities to become involved with us—including opportunities to influence the development of our products and services. Section 4.4 contains details about how we involve our customers and keep them informed about our work.

Education plays a significant role in technology transition. SEI education-related activities aim to improve software engineering practices and advance software engineering as a profession. These activities range from presenting seminars for executives to training trainers of software practitioners to influencing curriculum decisions at U.S. universities. See Section 4.5 for more on the role of education in technology transition.

Another key transition activity is providing guidance and advice on process improvement and technology transition. We work with organizations that are influential leaders in the software community, helping them to build and sustain continuous improvement of their software processes and their processes for adopting new technologies. Section 4.6 describes these ser-

vices of the SEI, demonstrating how we can help to develop an infrastructure where one is needed, open new channels for technology transition, and prepare organizations for the changes that accompany their transition and improvement activities.

We will know our mission has been accomplished when an engineering profession for software is well defined and widely accepted, and when there is a self-sustaining infrastructure for the software engineering community.

4.1 Technology Transition Models

At present the transition of software technology, like that for many other technologies, is ad hoc, unpredictable, lengthy, and expensive. The solution to this problem lies in understanding its nature: transition requires change in the behavior of individuals, change in the organizations where these individuals work, and more predictable and more rapid maturation of technology. Models—abstractions that represent approaches to solving particular technology transition problems—provide a way to unravel the complexity of transition. The SEI has developed or adapted several models that help us to think systematically about these problems and to develop an integrated strategy for technology transition.

Figure 4-1 illustrates a conceptual framework for viewing the transition process in terms of three interlocking life cycles. These are: research and development (including the creation of prototypes), new product development, and technology adoption and implementation. Together, these three life cycles cover technology development and transition from the inception of a technology until its retirement.

The SEI offers products and services that help to pull the life cycles together, thereby shortening the maturation time for software technologies. First we offer products and services that focus on promising concepts from research. Examples are surveys of new technology in terms of users' needs, conceptual frameworks, standard vocabularies for discussing and comparing alternative technologies, and evaluation methods for winnowing the possibilities and alternatives. For example, the capability maturity model (CMM) is a framework for process improvement; the risk taxonomy is a vocabulary for looking at sources of risk.

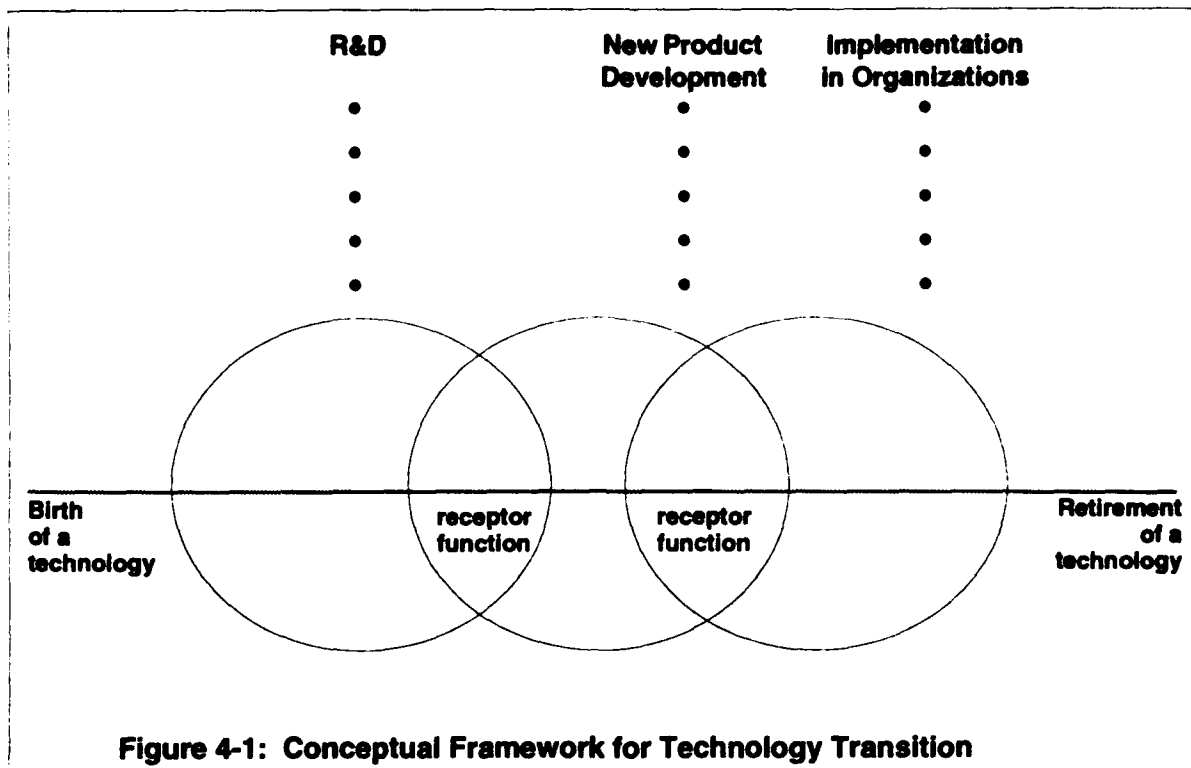
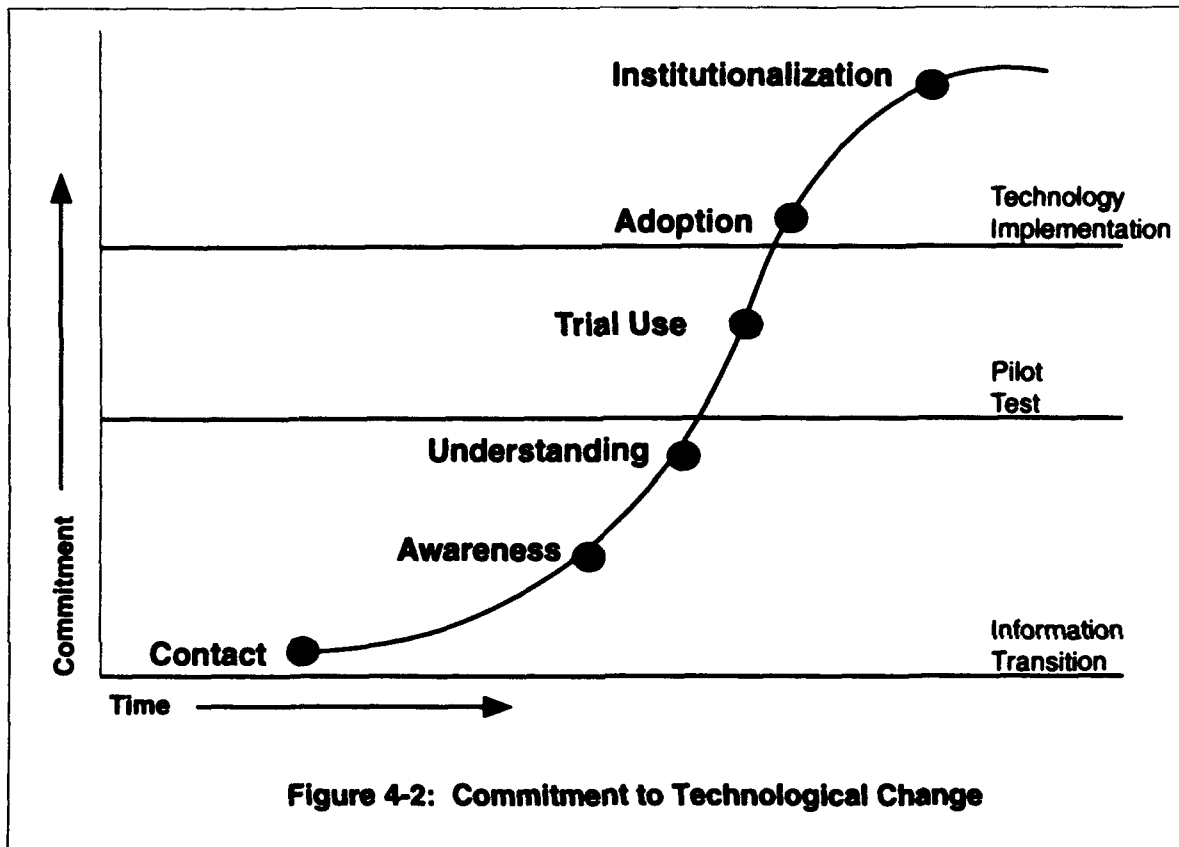


Figure 4-1: Conceptual Framework for Technology Transition

Next, the SEI offers products and services that help move technologies selected for advanced development into the distribution chain. The focus here is on technology producers and advocates (such as vendors of commercial off-the-shelf software). Products and services include evaluation benchmarks, usage scenarios, and various means for building community consensus regarding the purpose, form, and use of the technology. In some cases, services include identifying transition partners or channels for the technology and facilitating its transition. For example, the handbook on rate monotonic analysis (RMA) codifies the use of RMA in a variety of situations; the computer-aided software engineering (CASE) adoption guide and Ada adoption guides assist organizations in implementing a new technology.

Although Figure 4-1 appears conceptually simple, an inherent complexity becomes evident when we begin to apply it. Decisions must be made about when to combine what actions at what time and for whom, depending on the role key participants are playing.

One important model (Figure 4-2) that the SEI uses to help make those decisions is adapted from Conner [Conner 82]. This model shows how people and organizations commit to using a new software technology over time.



Each stage on the curve in Figure 4-2 represents a significant aspect of how commitment is made to a new technology.

Information Transition

In the first stage, people and organizations first learn of a technology. During the next two stages, they gain increasing awareness of the technology, its value, and its limitations. They develop an understanding of how a technology might be relevant to their specific needs.

These three stages are best addressed through information-intensive mechanisms such as education, newsletters, electronic bulletin boards, technical reports, short videotapes, journal and magazine articles, conferences, and demonstrations. These mechanisms enable users of a technology to come into contact with the technology and gain awareness of its purpose and application.

Examples of SEI transition activities and products for the contact, awareness, and understanding stages include: SEI seminars for executives (education); *Bridge* (news magazine); agreements with the Defense Technical Information Center (DTIC), National Technical Information Service (NTIS), and Research Access Inc. (RAI) to distribute SEI technical reports; and events such as the SEI Software Engineering Symposium, Conference on Software Engineering Education, and Risk Conference.

Pilot Test

If experiences during the contact, awareness, and understanding stages are positive, potential adopters for a technology may decide to try it out to determine how well it addresses their needs. Trial use then initiates the next stage. This and later stages are more labor-intensive for the adopters and for the people who support them.

Here, transition mechanisms include training to build skills, new or revised standards and procedures, use of new tools, apprenticeships, customer services, consulting, and perhaps even revision of reward systems. The payoff for this investment comes when positive results in trial use lead to broader use. Products and services that help organizations move to the next stage are those that define maturity stages and factors, along with yield estimators.

Examples of SEI support for the trial use stage include: SEI courses for practitioners, Hartstone benchmark prototype software, working with resident affiliates, and licensing SPA associates.

Technology Implementation

With adoption, the organization commits to critical path use of a technology. In institutionalization, the technology is the technology of choice for the problem area it addresses and for all candidate users.

To support the move into widespread use, products and services are needed to define and support migration paths from current practice to the new best practice. Examples are specialized adoption models and strategies; creation or identification of transition vehicles for the technologies; and handbooks and guides, which provide reference materials for daily use. This class of products and services can be described as transition efforts intended to move ever larger portions of the community ever higher on the adoption curve.

Examples of SEI products and services for organizations at the adoption and institutionalization stages include: curriculum dissemination directly to universities so that the next generation of software engineers is well prepared; use of the National Technological University as a transition partner; train-the-trainer courses; the *RMA Handbook* and Ada adoption guides; in-

volvement with Software Process Improvement Network (SPIN) groups that allow software engineering process groups (SEPGs) to learn from each other; and participation in standards development such as ISO 9000 and FutureBus+.

The model in Figure 4-2 reminds us of the magnitude of change new software technology can demand. For the SEI to achieve its mission, each individual in each organization and community must progress through several stages of commitment for each technology that is to be adopted. The challenge is great; the population addressed by the SEI transition effort exceeds 250,000 technical professionals, their managers, and the academic and continuing education community that serves these individuals. The SEI must find ways to leverage its resources.

The SEI extends and amplifies its impact by selecting products and services that can have the most far-reaching effect and by using the existing infrastructure, including the commercial infrastructure, to the greatest extent possible. The following sections in this chapter describe in more detail some of the ways the SEI does this.

4.1.1 Investigations into Technology Transition Models

Because software technology transition is one of the SEI core competencies, we conduct investigations that help us determine how to accomplish transition of software technology most effectively and how to create products that most effectively support our customers' improvement efforts. We are investigating approaches to software technology transition and developing a framework within which transition strategies can be selected and mechanisms chosen.

4.1.1.1 Five-Year Plan

4.1.1.1.1 Goals

The long-term goal of our investigations is to have a predictable, systematic, and replicable process of software engineering technology transition recognized and widely practiced within the SEPG community by 1999. We will know we are successful when the conceptual framework, vocabulary, models, and methods for planning and incorporating new technology in software organizations are taught and practiced as routinely as software project management, and when technology transition efforts are accomplished as predicted, on schedule, and within budget.

4.1.1.2 Strategy

- Assume a national leadership role in building consensus across the software engineering community on a conceptual framework, vocabulary, models, and methods for software technology transition.

- Establish a national presence for the SEI in the area of software technology transition models.
- Develop a model-based planning and implementation process for software technology transition. Refine the conceptual framework and vocabulary.
- Demonstrate the feasibility of a systematic, predictable, and replicable process—that is, an engineered approach to software technology transition—by analyzing SEI transition success stories in terms of the conceptual framework, vocabulary, models, and methods. Publish software transition case studies based on demonstration of transition methods developed or refined at the SEI.
- Educate executive and line managers and engineers regarding technology transition concepts for software engineering, along with vocabulary, models, and methods. Continue our development of an education and training program for line software managers, senior engineers, and members of software engineering process groups.
- Develop aids to support managers and engineers who must plan and implement software technology transition efforts. These may be paper-based or provided through software or other media. Initiate a joint venture with a vendor to beta test an aid for planning and decision making in software transition efforts.
- Establish an ongoing relationship with the National Technology Transfer Center, Council of Consortia, and related agencies and functions.

4.1.1.3 Potential Products

Workshop: Managing Software Technology Transition as a Project (2Q 1994). This one-day workshop provides concepts and basic skills for change agents within organizations who must introduce software technologies. It describes a conceptual framework and vocabulary for software technology transition, and transition models in the categories of people, organization and technology, within a framework of steps for transition planning. The workshop includes brief exercises and a case study.

Revised workshop: The Technology of Technology Transition (3Q 1994, tentative; depends on TO&P). This three-day workshop provides concepts and basic skills for technology developers who are preparing their technologies for transition. It describes a conceptual framework, models, and vocabulary for software technology transition, and addresses approaches to product development and receptor organizations that can enhance a technology's transition potential.

Special report on prototype aid beta test results (3Q 1994). The functional prototype (key functions) of the aid for change agents who must introduce software technologies into organizations will be evaluated through a series of beta tests. This report will mark the completion of this work and will describe results to change agents.

SEI technical report: Conceptual Framework for Software Technology Transition, v. 2 (4Q 1994). This report, for change agents, technology developers, software managers, and software engineering educators, will provide an updated and extended version of the technical report of the same title delivered in 1993. It will contain the conceptual framework, vocabulary, and basic set of transition models for software technology transition.

Lessons learned report (4Q 1994). This report, for change agents, technology developers, software managers, and software engineering educators, will present a detailed description and analysis of experience in applying the software technology transition conceptual framework and models to a specific transition situation for an SEI technology.

Special report on results of beta test of software aid prototype (1995-1998). The final prototype (all functions) of the aid for change agents who must introduce software technologies into organizations will be evaluated through a beta test. This report will mark the completion of this work, which is a joint effort, and will describe results to change agents.

Case studies (1995-1998). Formal case studies covering a range of software technology transition situations will be conducted and documented. These case studies will provide opportunity for change agents, technology developers, managers, and software engineering educators to learn about transition through the experience of others. The case studies will provide analyses of both successes and limitations of transition of specific software technologies.

Education and training programs (1995-1998). A curriculum in software technology transition will be designed and developed. It will address the need for both education and training in software technology transition concepts, terminology, models, planning and management. Ultimately these education and training programs will be licensed.

4.1.1.4 One-Year Plan

4.1.1.4.1 Context

In 1993, we completed the technical report, *A Case Study of the Transition of Rate Monotonic Analysis*. This report provides an example for technology developers who must plan the transition of their technologies. We also developed a conceptual framework for software technology transition and published a short, preliminary version in a *Bridge* article and, later, a technical report entitled *Software Technology Transition: A Conceptual Framework*. We developed, with the Universidad Politecnica de Madrid, a paper prototype of an aid for those who must introduce new software technologies into organizations. We also developed and delivered a tutorial, *Managing Software Technology Transition as a Project* (at the SEPG National Meeting and the SEI Symposium); this tutorial will serve as a prototype for the workshop that is a 1994 deliverable.

4.1.1.4.2 Core Funded Activities

In this activity, we work toward describing the overall transition process flow from the inception of a technology to its retirement. For each of three major arenas of technology transition—R&D, new product development, and implementation of technology in organizations—there is a set of tasks that must be accomplished, specific terminology that applies, and one or more disciplinary bases to draw from. Because the SEI transition efforts range across all three arenas, it is important to articulate clearly each arena and how each interacts with the other. This can provide the SEI and its constituents a common framework within which to understand technology transition activities and relationships. Our relationship with the Council of Consortia will provide input, as will continuing attendance at important conferences and meetings within the technology transfer community, and literature reviews. We will build on and extend the work in version 1 of the SEI technical report, *Software Technology Transition: A Conceptual Framework*.

The conceptual framework, vocabulary, models and methods developed will then be used to plan and implement the transition of an SEI technology. We have plans to explore an effort in process definition technology with other SEI staff. Because software technology transition can be described as a process, this work will also help refine the overall transition flow mentioned above. It should also provide input as we refine the aid for change agents in the activity described next.

We will continue our work to develop a software-based aid for change agents who must introduce software technology into their organizations by beta testing a prototype in several organizations. An effective transition plan usually combines a number of mechanisms such as negotiation, briefings, training, electronic mail and bulletin boards, marketing materials, and documentation of various types. Most software practitioners and managers have experience with these mechanisms but have only limited experience in choosing and combining them for use in a transition situation. This aid will provide transition problem analysis and technology implementation planning support. The aid is also a vehicle for accumulating, in an organized and focused way, our experience and knowledge in software technology transition.

Because the aid will take some time to develop, we are working to develop a one-day hands-on workshop for change agents, either managers or practitioners. This workshop integrates the results of the three activities just described in a practical short course on how to plan and implement transition in an organization.

4.1.1.4.3 TO&P Funded Activities

The Air Force has expressed interest in developing the concept of "technology broker" so that technology maturation can be managed more effectively. We will support an Air Force working group on software technology transition in evolving this concept. As part of this effort, we will

build on the conceptual framework for software technology transition already developed at the SEI by tailoring it and related terminology for Air Force use. This work will contribute to evolving version 2 of the framework as discussed above.

The workshop described above, Managing Technology Transition as a Project, will be tailored for Air Force change agents and presented as resources allow and in the context of an overall plan for supporting the Air Force that is now in preparation. (Negotiations are underway with the Embedded Computer Resources Support Improvement Program (ESIP) for several years of support in this area.)

We will revise the workshop, The Technology of Technology Transition, that has been taught for several years to Air Force change agents and technology transition managers. The goal of the revision is consistency with the conceptual framework described above and improvements in response to comments received to date. The workshop will continue to provide concepts and basic skills for technology developers who are preparing their technologies for transition.

We will continue our support of technology receptor organizations such as Air Mobility Command and Air Combat Command. This work involves assistance in planning and implementing pilot uses of software technology. This work can be combined in part with testing of the prototype aid to change agents.

We will provide other technology transition training and education, including the Managing Technological Change and the Consulting Skills workshops.

Air Force-specific software technology transition case studies will be considered and developed if appropriate as vehicles of surrogate experience in software technology transition for both practitioners and managers.

1995-1998

The final prototype (all functions) of the aid for change agents will be evaluated through beta testing in a number of organizations. If results show that the aid is promising, partners will be sought for creation of a commercial tool.

We will conduct and document formal case studies covering a range of software technology transition situations. We have identified a set of typical software transition situations, and case studies for these will enable change agents, technology developers, managers, and software engineering educators to gain surrogate learning about transition. The case studies will provide analyses of both successes and limitations of a range of transition situations, to allow emulation of success and avoidance of problem areas.

The workshops currently available or under development for 1994 will be reviewed and evolved into a curriculum in software technology transition for managers and practitioners. The curriculum will address the need for both education and training in software technology transition concepts, terminology, models, planning and management, and will supplement the material contained in the aid for change agents.

4.1.1.4.4 Activities Funded from Sources Other than Core and TO&P

The SEI will work with the Technology Transfer and Best Practices Working Groups of the Council of Consortia to develop and alpha test a prototype support tool for performing technology transition planning.

4.2 Products and Services

To achieve the leverage required to have a significant impact on the state of the practice, the SEI designs products and services to facilitate and expedite software technology transition. This section describes our products and services in terms of their form and the segment of the software engineering community for which they are intended.

To support the effort of the software community to improve its practice, we offer these varied products and services:

- **Courses:** a structured set of learning experiences designed to change learner behavior based on prespecified performance objectives. (See Section 4.5 for a description of the role education plays in technology transition.)
- **Events:** an SEI-sponsored or co-sponsored activity, such as a meeting, workshop, symposium, or conference, that is designed to promote information exchange between us and our customers and among the members of the software community.
- **Publications:** a standard SEI document, which may be published by CMU or published by a transition partner, such as Addison-Wesley, Springer-Verlag, or Kluwer Academic Publishers.
- **Prototype software:** code in source or object form (or both) that implements a process, method, or tool.
- **Videotapes:** instructional or informational programs, available as stand-alone products such as those in the SEI Technology Series. (Videotapes may also be a part of a course or educational materials.)
- **Guidance and advice:** our primary form of service to practitioners, managers, educators, or project teams, to help them adopt improved practices in their organization. SEI activities include working directly with customer organizations that are influential leaders in the software

community, assisting professionals who play consulting roles in improvement efforts, and promoting the development of industry and government infrastructures that support the adoption of improved practices. (See Section 4.6 for more detail about the role services play in technology transition.)

To more effectively serve our customers, we have divided products and services into three product categories, each of which addresses the problems of a segment of the software community:

1. Those that help senior *managers* do a better job of leading software organizations to build higher quality software and to do so more productively.
2. Those that help *software practitioners* produce and maintain better software.
3. Those that enable *educators* to improve software engineering education and training.

Although we draw from all the technical activities (described in Chapter 3) and the software community at large for products and services, there are some natural associations based on the needs of the customers. For managers, most products and services come from the areas of software process and software risk management. For practitioners, most come from software methods and tools and real-time systems. Because of the key role education plays in transition (see Section 4.5), products and services for educators cut across all these areas.

In 1992 we developed a package of materials that describes currently available products and services, arranged into product categories, so our customers can more easily learn what we have to offer. This package is being updated for 1993 (a summary appears in Appendix A). Our objective is to offer a cohesive set of products and services that help managers, practitioners, and educators improve software engineering practices. A qualitative measure of success for the SEI transition activities is the robustness and usefulness of our products.

4.2.1 Manager Products

Managers in the software community are concerned that systems meet performance requirements and are developed on schedule and within cost. Software managers typically function in crisis-driven environments that are characterized by ad hoc processes, a reliance on individual talents, a lack of visibility and predictability, and little insight into the impact of requirements changes on the systems they are building. Managers in software development organizations face problems associated with the process of building and maintaining software. Managers in the acquisition business face problems such as evaluating potential contractors, identifying risks inherent in managing software contracts, and managing those risks. Managers address these issues through the application of disciplined and systematic methods such as the CMM, risk management, and management of technological change. They need new

and different ways to determine the risks inherent in their organizations and programs, to raise the maturity level of their organizations, and to decide what level of investment and priority should be given to an improvement program.

SEI products are designed to help managers solve their problems through risk management, software process improvement, and technology adoption in their organizations. The primary focus of these products is on (1) improving the software process by raising the maturity level of these organizations to one where an organizational capability exists and an acceptable level of visibility and predictability is achieved; (2) identifying and mitigating organizational and program risks; and (3) facilitating adoption and institutionalization of new technologies to improve software quality and productivity into organizations.

4.2.2 Practitioner Products

Practitioners involved in software development and maintenance are concerned about productivity, quality, and dependability. They address these by disciplined application of methods and tools to predict and control the behavior of their software systems. They need new paradigms and models to predictably create systems that possess essential characteristics such as fault tolerance, interoperability, and modifiability. Increasing the predictability of the engineering methods and tools will dramatically increase developers' productivity. Practitioners also need to ensure, throughout the system life cycle, that the specifications they write and the software they build will meet their customers' needs. And because the software systems are complex, practitioners need to identify integrated technologies and tools that support their methods, that are right for transition, and that can be adopted within their organization.

Products for practitioners provide software engineering methods and tools that address a wide range of software issues, including performance, safety, fault tolerance, interoperability, re-use, reengineering, and modifiability. These products help practitioners improve their effectiveness and efficiency in engineering and reengineering large software-intensive systems and improve their ability to predict and control the development process and the quality of the systems they develop.

There are three primary thrusts for products that document and automate process and product technologies:

1. Methods and tools using current best practice applicable to all domains.
2. Methods and tools using current best practice and qualitative methods for specific application domains.
3. The development of new quantitative methods, such as RMA, for added predictive capability for specific system qualities.

These will help to improve the state of software practice and raise the maturity of the software engineering profession.

4.2.3 Educator Products

Educators face the challenge of preparing people to become productive software engineers and improving their performance after they are in the field. This involves understanding what constitutes a good software engineer (knowledge, skills, and qualities) and integrating this knowledge with efforts to improve (1) the educational processes and practices that identify and prepare software engineers, and (2) the organizational processes and practices that support software engineers and enable them to continuously improve throughout their career.

University educators must define academic programs that establish the discipline of software engineering at both the graduate and the undergraduate level to adequately prepare new software engineers entering the work force. Educators and trainers of current professionals must meet both immediate and long-term needs. They must provide training in new processes, methods, and tools, and also find ways to provide for life-long learning in a field that is rapidly changing to ensure that software engineers avoid obsolescence throughout their career.

Products for educators are designed to promote and foster software engineering as a leading discipline and viable career path for software professionals through life-long learning opportunities. The technical foundation for many of these products is the evolving SEI model curricula for graduate and undergraduate degree programs in software engineering. Courses and other educational materials aid educators who teach in these programs. Courses for continuing education and training integrate topics and best practices relating to software process, methods and tools, real-time systems, and risk. For educators preparing future generations of software engineers in academia and for those keeping current practitioners up to date in industry and government, the products emphasize the teaching of software engineering concepts and principles that pave the way for effective skill-based, experiential, and just-in-time learning.

4.2.4 Summary of SEI Product Plans

The following tables summarize the products planned for 1994 and potential products that may be developed in 1995-1998. All are subject to SEI internal review and quality assurance activities before we make them widely available.

4.2.4.1 Products Planned for 1994

| | Product | Manager | Producer | Editor | Reference |
|---------|--|---------|----------|--------|----------------------------|
| PROCESS | CMM v 1.1 Training | ▼ | ▼ | ▼ | Section 3.2.1.3 page 35 |
| | CMM Human Resource Maturity Enhancement | ▼ | ▼ | ▼ | Section 3.2.1.3 page 35 |
| | SPA Training (upgrade) | ▼ | ▼ | | Section 3.2.1.3 page 36 |
| | SPA Method (upgrade) | ▼ | | | Section 3.2.1.3 page 36 |
| | SPA Method (upgrade) for Levels 4 and 5 | ▼ | | | Section 3.2.1.3 page 36 |
| | SPA Tailoring Techniques | ▼ | | | Section 3.2.1.3 page 36 |
| | SCE Training (upgrade) | ▼ | ▼ | | Section 3.2.1.3 page 36 |
| | SCE Method (upgrade) | ▼ | | | Section 3.2.1.3 page 37 |
| | Process Definition Training | ▼ | ▼ | | Section 3.2.1.3 page 37 |
| | Measurement Workshop | ▼ | ▼ | | Section 3.2.1.3 page 38 |
| | Cost Estimating Techniques | ▼ | ▼ | | Section 3.2.1.3 page 38 |
| | Measurement Definition Checklists & Frameworks | ▼ | ▼ | | Section 3.2.1.3 page 38 |
| | Instant Profile | ▼ | ▼ | | Section 3.2.1.3 page 39 |
| | Process Value Method Pilot | ▼ | ▼ | | Section 3.2.1.3 page 39 |

4.2.4.1 Products Planned for 1994 (continued)

| | Product | Manager | Engineer | Editor | Summary Page Number |
|------|---|---------|----------|--------|----------------------------|
| RISK | SEI Software Risk Conference | ▼ | ▼ | ▼ | Section 3.3.1.3 page 52 |
| | Risk Identification Training Course | ▼ | ▼ | | Section 3.3.1.3 page 53 |
| | Software Risk Evaluation (SRE) Handbook | ▼ | ▼ | | Section 3.3.1.3 page 53 |
| | Independent Risk Assessment | ▼ | | | Section 3.3.1.3 page 54 |
| | Team Risk Management | ▼ | | | Section 3.3.1.3 page 55 |
| | Team Risk Management Tutorial | ▼ | ▼ | | Section 3.3.1.3 page 55 |

| | Product | Manager | Engineer | Editor | Summary Page Number |
|-------------------|--|---------|----------|--------|----------------------------|
| METHODS AND TOOLS | Tailorable Model-Based Software Engineering Course Collection | ▼ | ▼ | ▼ | Section 3.4.1.3 page 75 |
| | Advanced Multimedia Organizer for Requirements Elicitation (AMORE) | ▼ | ▼ | | Section 3.4.1.3 page 75 |
| | Guide to Best Practice in SEEs: Edition 1 | ▼ | ▼ | | Section 3.4.1.3 page 76 |
| | Annual SEI Software Engineering Techniques Workshop: Reengineering | ▼ | ▼ | | Section 3.4.1.3 page 77 |

4.2.4.1 Products Planned for 1994 (continued)

| | Product | Manager | Practitioner | Editor | Review Manager/Editor |
|-------------------|--|---------|--------------|--------|----------------------------|
| REAL-TIME SYSTEMS | Open Systems Standard for Real-Time Communications (draft) | | ▼ | | Section 3.5.1.3 page 91 |
| | Systems Fault Tolerant Handbook | | ▼ | | Section 3.5.1.3 page 91 |
| | Airborne Radar Study Reports | | ▼ | | Section 3.5.1.3 page 91 |
| | STRICOM Reports | | ▼ | | Section 3.5.1.3 page 92 |
| | Real-Time Simulators Guidebook | | ▼ | | Section 3.5.1.3 page 92 |
| | Software Developer's Handbook (draft) | | ▼ | | Section 3.5.1.3 page 93 |
| | Systems Fault Tolerant Technology Exchanges | ▼ | ▼ | | Section 3.5.1.3 page 94 |
| | Software Technology Awareness Videos | ▼ | ▼ | | Section 3.5.1.3 page 94 |
| | Software Developer's Forum | ▼ | ▼ | | Section 3.5.1.3 page 94 |

| | Product | Manager | Practitioner | Editor | Review Manager/Editor |
|-------------------|---|---------|--------------|--------|-----------------------------|
| TRANSITION MODELS | Workshop: Managing Software Technology Transition as a Project | ▼ | ▼ | | Section 4.1.1.3 page 109 |
| | Special report on prototype aid beta test results | ▼ | ▼ | | Section 4.1.1.3 page 109 |
| | SEI technical report : Conceptual Framework for Software Technology Transition, version 2 | ▼ | ▼ | | Section 4.1.1.3 page 110 |
| | Lessons learned report | ▼ | ▼ | ▼ | Section 4.1.1.3 page 110 |

4.2.4.1 Products Planned for 1994 (continued)

| | Product | Manager | Project Lead | Engineer | Product Plan Page |
|------------------------------------|---|---------|--------------|----------|-------------------------------|
| EDUCATION IN TECHNOLOGY TRANSITION | Directory of industry and university consortia | ▼ | ▼ | ▼ | Section 4.5.1.1.3 page 134 |
| | Course on real-time software design and development | | | ▼ | Section 4.5.1.1.3 page 135 |
| | Revision of academic course Software Construction with Ada | | | ▼ | Section 4.5.1.1.3 page 135 |
| | Update of academic course Software Design | | | ▼ | Section 4.5.1.1.3 page 135 |
| | Update of academic course Software Verification and Validation | | | ▼ | Section 4.5.1.1.3 page 135 |
| | Two to three courses that support key process areas at levels 2 and 3 of the SEI capability model (CMM) | ▼ | ▼ | ▼ | Section 4.5.1.1.3 page 135 |
| | Faculty development seminars | | | ▼ | Section 4.5.1.1.3 page 135 |
| | Conference on Software Engineering Education | | | ▼ | Section 4.5.2.1.3 page 138 |
| | 1994 SEI Report on Software Engineering Education | ▼ | ▼ | ▼ | Section 4.5.2.1.3 page 139 |
| | Educational materials | | | ▼ | Section 4.5.2.1.3 page 139 |

| | Product | Manager | Project Lead | Engineer | Product Plan Page |
|----------|---------------|---------|--------------|----------|---------------------------|
| SEI-WIDE | Visitors Day | ▼ | ▼ | ▼ | Section 4.4.1 page 126 |
| | SEI Symposium | ▼ | ▼ | ▼ | Section 4.4.7 page 130 |

4.2.4.2 Potential Products for 1995-1998

| | Product | Manager | Practitioner | Educator | Reference (Section and page number) |
|---------|---|---------|--------------|----------|--|
| PROCESS | Process Definition Train-the-Trainer (1995) | ▼ | | ▼ | Section 3.2.1.3 page 38 |
| | Process Value Method (1995) | ▼ | ▼ | | Section 3.2.1.3 page 39 |
| | CMM V 2.0 (1996-1997) | ▼ | ▼ | ▼ | Section 3.2.1.3 page 35 |
| | CMM Levels 4&5 Enhancements (1996-1997) | ▼ | ▼ | ▼ | Section 3.2.1.3 page 35 |
| | CMM for Small Organizations (1996-1997) | ▼ | ▼ | | Section 3.2.1.3 page 35 |
| | CMM Validation Study (1996-1997) | ▼ | ▼ | | Section 3.2.1.3 page 39 |

| | Product | Manager | Practitioner | Educator | Reference (Section and page number) |
|------|---|---------|--------------|----------|--|
| RISK | SEI Software Risk Conference (annual) | ▼ | ▼ | ▼ | Section 3.3.1.3 page 52 |
| | Tailorable Taxonomy-Based Questionnaire (1995) | ▼ | ▼ | | Section 3.3.1.3 page 53 |
| | Risk Analysis Training Course (1995) | ▼ | ▼ | | Section 3.3.1.3 page 53 |
| | Software Risk Evaluation (SRE) Train-the- Trainer (1995) | | | ▼ | Section 3.3.1.3 page 54 |
| | Predictive Decision Tool (1995) | ▼ | | | Section 3.3.1.3 page 54 |
| | Team Risk Management User's Guide (1995) | ▼ | | | Section 3.3.1.3 page 55 |
| | Risk Data Repository (1996) | ▼ | ▼ | ▼ | Section 3.3.1.3 page 53 |
| | Software Risk Capability Improvement Guide (1996) | ▼ | ▼ | | Section 3.3.1.3 page 53 |
| | Program Manager's Risk Indicators and Alerts Tool (1996) | ▼ | | | Section 3.3.1.3 page 55 |
| | Program Manager's Assistant (1997) | ▼ | | | Section 3.3.1.3 page 55 |

4.2.4.2 Potential Products for 1995-1998 (continued)

| | Product | Manager | Practitioner | Educator | Formative Information Uses |
|-------------------|---|---------|--------------|----------|----------------------------|
| METHODS AND TOOLS | Guide to Best Model-Based Software Engineering Practice: Edition 1 (1995) | ▼ | ▼ | | Section 3.4.1.3 page 75 |
| | Roadmap for Environment Technology (1995) | ▼ | ▼ | | Section 3.4.1.3 page 76 |
| | Guide to Best Reengineering Practice: Edition 1 (1995) | ▼ | ▼ | | Section 3.4.1.3 page 77 |
| | Guide to Best Practice in System Understanding: Edition 1 (1996) | ▼ | ▼ | | Section 3.4.1.3 page 76 |
| | Guide to Best Practice in SEEs: Edition 2 (1997) | ▼ | ▼ | | Section 3.4.1.3 page 76 |
| | Guide to Best Model-Based Software Engineering Practice: Edition 2 (1998) | ▼ | ▼ | | Section 3.4.1.3 page 75 |
| | Intelligent SEEs (1998) | ▼ | ▼ | ▼ | Section 3.4.1.3 page 76 |

| | Product | Manager | Practitioner | Educator | Formative Information Uses |
|-------------------|--|---------|--------------|----------|----------------------------|
| REAL-TIME SYSTEMS | Software Developer's Handbook (annual) | | ▼ | | Section 3.5.1.3 page 93 |
| | Software Developer's Forum (annual) | ▼ | ▼ | | Section 3.5.1.3 page 94 |
| | Systems Fault Tolerant Technology Exchanges (bi-annual) | ▼ | ▼ | | Section 3.5.1.3 page 94 |
| | Software Technology Awareness Videos (multiple per year) | ▼ | ▼ | | Section 3.5.1.3 page 94 |
| | Open Systems Architecture Handbook (1995) | ▼ | ▼ | | Section 3.5.1.3 page 91 |
| | Open Systems Standard for Real-Time Communications (1995) | | ▼ | | Section 3.5.1.3 page 91 |
| | Open Systems Standard for High Performance Networks (1996) | | ▼ | | Section 3.5.1.3 page 91 |
| | Core Architecture for Flight Simulators (1996) | | ▼ | | Section 3.5.1.3 page 92 |

4.2.4.2 Potential Products for 1995-1998 (continued)

| | Product | Manager | Practitioner | Evaluator | Formal information source |
|-------------------|--|---------|--------------|-----------|-----------------------------|
| TRANSITION MODELS | Special report on results of beta test of software aid prototype (1995-1998) | ▼ | ▼ | | Section 4.1.1.3 page 110 |
| | Case studies (1995-1998) | ▼ | ▼ | ▼ | Section 4.1.1.3 page 110 |
| | Education and training programs (1995-1998) | | | ▼ | Section 4.1.1.3 page 110 |

| | Product | Manager | Practitioner | Evaluator | Formal information source |
|------------------------------------|---|---------|--------------|-----------|-------------------------------|
| EDUCATION IN TECHNOLOGY TRANSITION | Conference on Software Engineering Education (held annually) | | | ▼ | Section 4.5.2.1.3 page 138 |
| | Faculty development seminars (1994-1998) | | | ▼ | Section 4.5.1.1.3 page 135 |
| | Two continuing education courses relating to reuse (1995) | ▼ | ▼ | ▼ | Section 4.5.1.1.3 page 135 |
| | Continuing education courses that support CMM key process areas and/or special topics (1995-1998) | ▼ | ▼ | ▼ | Section 4.5.1.1.3 page 139 |
| | Educational materials (1995-1998) | | | ▼ | Section 4.5.2.1.3 page 139 |
| | Revised model curricula (1995-1998) | | | ▼ | Section 4.5.2.1.3 page 139 |

4.2.4.2 Potential Products for 1995-1998 (continued)

| | Product | Manager | Practitioner | Educator | For more information, see |
|----------|-------------------------------|---------|--------------|----------|---------------------------|
| SEI-WIDE | Visitors Day (three per year) | ▼ | ▼ | ▼ | Section 4.4.1 page 126 |
| | SEI Symposium (annual) | ▼ | ▼ | ▼ | Section 4.4.7 page 130 |

4.3 Transition Partners

Transition partners provide high leverage in SEI efforts to accelerate the reduction to practice and promulgate the use of modern software engineering techniques and methods. By taking advantage of existing delivery structures, transition partnerships save the time and expense of creating new infrastructure exclusive to the SEI. The most successful transition partners are those who are already active in the market segment to which they intend to target the SEI product and who have sufficient resources and motivation to actively promote and support their SEI-related effort.

In selecting transition partners, the SEI assesses their potential for success and their appropriateness for the role of transitioning SEI-related material. Some of the factors that are considered include

- Composition of their current customer base
- Technical experience and competence
- Nature of mutually acceptable terms of partnership
- Willingness to enter into a non-exclusive arrangement

There is a variety of transition partner roles and approaches. Organizations like NTIS and DTIC represent one type of transition partner. The role of these organizations is purely information dissemination. They are not generally considered commercial organizations, and they do not require ongoing SEI support. They do not have resource impact on the institute, either positive or negative. DTIC is the organization with which the SEI has had its longest running transition partner arrangement. All technical reports and other documents that have been approved for public release by the Joint Program Office are available to government organizations through DTIC.

Another type of transition partner is represented by the software process assessment (SPA) associates, some of whom work closely with the SEI as the product matures and then deliver it. This type of partner acquires and adds value to advanced technology or management techniques that were developed by the SEI and that the partner believes can help their customers enhance the effectiveness of their work. The value added by this type of partner is primarily technical and, in some cases, consultative in nature. Most often the partner is already in a software-related business. From the SEI point of view, this category of partner requires continuing contact and support over time.

Ideally, these transition partners become involved with the SEI very early in the development process and help to identify potential customer needs and define corresponding product features and benefits. Later in the development cycle, they collaborate in piloting the new product or technology. Thus, in this case transition partners can become development partners as well, which provides further leverage for the DoD investment.

More often, as in the case of the SPA associates, the SEI will have done much of the market analysis, product design, and early development work before transition partners are identified. The advantage of this approach is that multiple transition partners participate in the final stages of product development, and the resulting synergism allows broader coverage in a shorter period of time than would otherwise be the case.

A third type of transition partner is represented by organizations like National Technological University (NTU) and RAI. NTU broadcasts SEI courses by satellite, and RAI distributes SEI documents. These organizations are more typical of partners in today's economic environment. Their objective is to acquire the rights to distribute SEI products, for which they are willing to pay royalties. Primary contributions of this category of partner are marketing and scope of delivery or coverage. Most often this type of partner is already in some form of delivery or distribution business but not necessarily in a software-related field. From the SEI point of view, this type of partner requires very little ongoing support.

This type of transition partner becomes involved only after the SEI has produced a mature product. In most cases, the SEI may have already been delivering the product and is considering a partnership in order to relieve customer demand on the SEI's fragile delivery infrastructure. In these cases, the SEI builds a business case for the transition partner to aggressively pursue the transition of the SEI product.

National Defense University is a good example of how the SEI uses the existing infrastructure to reach large numbers of people. This transition partner has incorporated a series of SEI courses into their curriculum with only modest changes to the way the courses were originally designed to be used. The partnership makes it possible for our courses to reach a large government audience without a large investment on the part of either partner.

Appendix C contains a list of our current transition partners.

4.4 Customer Involvement

The SEI has developed a range of relationships, from those that satisfy customer needs for general information about the SEI and its technical program to those that involve collaboration and exchange of technology. The SEI provides opportunities for customers to participate in technical interchange meetings, workshops, conferences, and educational offerings as well as the acquisition and co-development of specific products and services. The ways in which our customers can work with the SEI are described below.

4.4.1 Customer Inquiry/Response

This service is provided through the SEI information line: (412) 268-5800; Internet e-mail: customer-relations@sei.cmu.edu; and FAX: (412) 268-5758. The customer information specialists who provide this service are fully prepared to answer any question of a general nature about the SEI, to mail pertinent descriptive materials, and to follow up with members of the SEI technical staff to provide more detailed information. The SEI provides this service during normal working hours, handling up to 200 requests per week. We collect statistics and maintain a database reflecting the character of the inquiry/response traffic. This information is often used by others at the SEI to contact and respond to our customer community.

Another way of reaching customers is through Visitor's Day, which is hosted by the SEI three times a year to familiarize software managers, practitioners, and educators with the SEI and its activities. Members of the SEI technical staff give presentations on the technical program, and SEI products and services.

4.4.2 Subscriber Program

The Subscriber Program is an effective way for an individual to stay informed about SEI activities. Participants receive mailings that keep them up to date on SEI events, course offerings, work in progress, new products, and new initiatives. Anyone with a United States mailing address is eligible to subscribe. A fee of \$100 (as of January 1993) has been established to help

offset costs of delivery. The fee covers an entire year from the date that the subscription is activated. It applies to industry and academia; government customers receive the same benefits at no cost by controlled distribution.

SEI subscribers currently receive the following:

- The quarterly *Bridge* magazine
- The annual *Technical Review*
- Discounts on technical reports
- A substantial discount at the annual SEI Software Engineering Symposium
- Early notification of SEI events

4.4.3 Resident Affiliate Program

The Resident Affiliate Program provides the opportunity for experienced technical personnel from government, industry, and academic organizations to participate in SEI projects. Resident affiliates contribute both as software engineers and as application domain experts, providing a valuable and practical perspective. They help us understand our customers' needs by providing information about their home organizations that helps us understand the organizational and technical contexts in which they practice software development.

The sponsoring organizations benefit by participating in technical activities that might not be possible in their own organizations, by obtaining early the results of SEI technical activities, and by having access to SEI people, projects, and other resident affiliates. The resident affiliate benefits from working in a different technical context, from participating in the many workshops and other activities at the SEI and in the larger CMU community, and from interacting with colleagues from different professional, technical, and organizational backgrounds. The SEI also benefits because it obtains experience, expertise, and additional insight into the software engineering community.

Resident affiliates work on site at the SEI for a negotiated period, usually 6 to 24 months; they may spend half to full time at the SEI. They devote approximately 80 percent of their time to an SEI technical project and the remaining time to technology transfer and liaison activities back to their home organizations. Resident affiliates are treated as integral members of the SEI staff.

Organizations that have participated in the Resident Affiliates Program are listed in Appendix B.

4.4.4 Advisory Boards/Working Groups

From time to time, the SEI identifies the need for a customer advisory board or working group to provide customer guidance on current activities and future plans, and to perform technical reviews of products. Members are selected through a screening process using project-defined criteria intended to populate the board or group with a mix of technical professionals who can help satisfy SEI technical objectives. The current advisory boards and working groups include the following:

- Software Process Program Advisory Board
- Process Definition Advisory Group
- Measurement Steering Committee
- Software Metrics Definition Working Group
- Software Acquisition Metrics Working Group
- Software Action Plan Measurement Team
- CMM Advisory Board
- CMM Correspondence Group
- CMM Review Group
- Software Capability Evaluation (SCE) Advisory Board
- SCE Review Group
- Software Dependability Working Group
- Risk Taxonomy Users Group

4.4.5 Software Process Improvement Network

In September 1992, the SEI agreed to serve as coordinator for the emerging software process improvement network (SPIN). The purpose of the SPIN groups is to satisfy customer needs for a practical forum for exchanging ideas, information, and mutual support on the subject of software process improvement. The primary role of the SEI is to disseminate information from existing SPIN organizations to groups of people in common geographical locations who are interested in starting new SPIN groups. The SEI maintains a directory of all currently active SPINs and points of contact in areas where interest has been expressed in forming a new SPIN. Active SPIN organizations have been formed in Washington, D.C., Irvine, California (serving Orange and Los Angeles counties), Dallas, Austin, Boston, and Seattle. New SPINs are emerging in Boulder, St. Louis, and Hoboken (serving northern New Jersey). Future services to be provided by the SEI include a SPIN newsletter and SPIN start-up kit.

The Software Process Improvement Network provides significant leverage for transition. The SEI interacts with SPIN groups, each member of which represents one or more SEPG. Each SEPG, in turn, represents a corporate software engineering staff often measured in the hundreds. Figure 4-3 illustrates this one-to-many effect.

4.4.6 Technical and Strategic Partnership Programs

The technical and strategic partnership programs are intended to create well-defined and well-managed relationships with the community of industry customers. Through these partnerships, the SEI has access to an industry constituency that can provide input to the SEI technical program; advance the maturity and accelerate the development of SEI technology, products or services; and provide in-kind and direct funding resources.

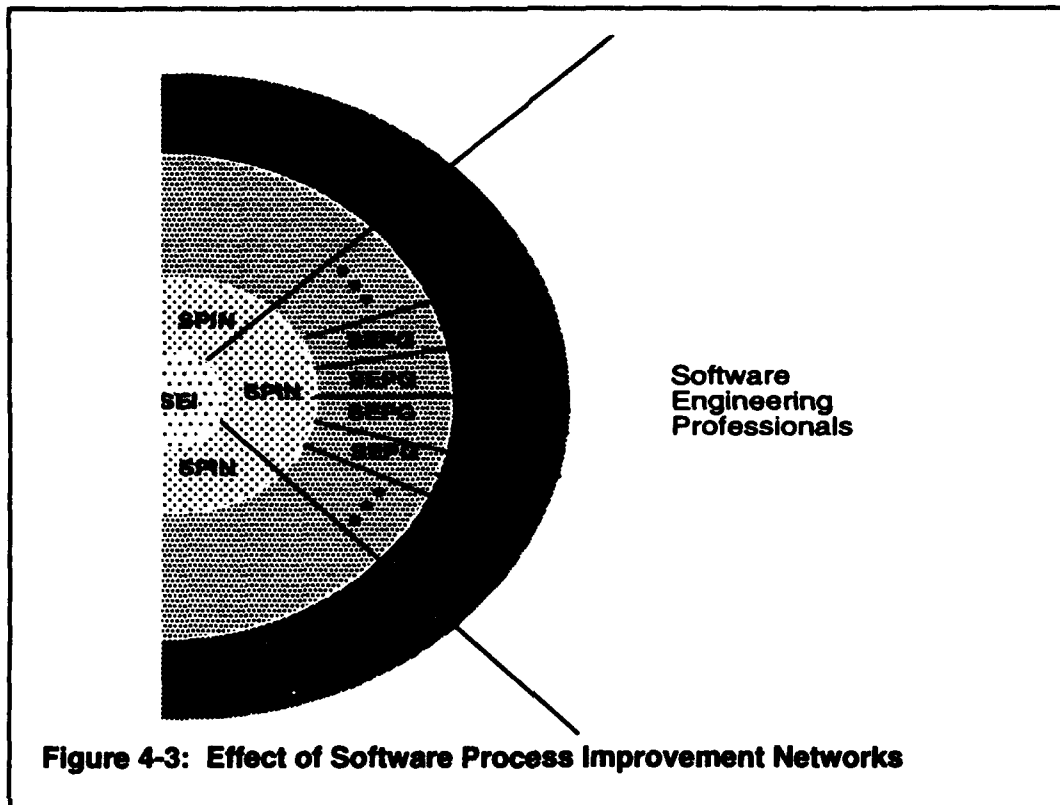
4.4.6.1 Technical Partnerships

Technical partnerships are formed for a fixed duration and involve well-defined areas of collaboration with a single SEI technical activity. Current examples include co-development of products of mutual interest (for example, roadmaps, field guides, handbooks, and training courses), early product review, and pilot/field testing of new products or processes. Technical partnerships are initiated by mutual agreement and are negotiated between the project and the potential partner with the intent of exchanging value of mutual benefit.

4.4.6.2 Strategic Partnerships

Strategic partnerships are long-term, collaborative relationships between the SEI and selected industry partners. These partnerships are characterized by mutual statements of strategic intent and goals. The strategic relationship is realized by executing multiple technical partnership agreements, as described above.

Candidate strategic partners have demonstrated their commitment to the SEI mission and vision and the transition of SEI-developed approaches by virtue of their historical and current involvement with the SEI. In addition, they have demonstrated a strong commitment to continuous improvement in the quality of their own software products and processes. The SEI seeks partners who are recognized leaders in several market segments, who have an ability to execute technology transition roles, and who can contribute to the depth and breadth of the SEI technical program. Benefits for strategic partners include broader, and often more immediate, access to SEI products, services, and technical staff; an opportunity to have input into SEI activities; and early access to products being co-developed, including products that may have been difficult to develop in a timely way without the benefit of collaboration.



4.4.7 SEI Software Engineering Symposium

This symposium is a major SEI event and perhaps our most visible technology transition activity. Held annually, it is the primary forum for presenting SEI work to the U.S. software engineering community. The symposium program includes a significant number of presentations from industry and government customers whose technology results relate to SEI work in progress.

This year's theme—The Business of Software Engineering: The Competitive Edge—reflects the Clinton/Gore conversion/dual-use agenda and the resulting expansion in the ARPA role to implement that agenda. It also extends the community traditionally served by the SEI to include the full range of defense and commercial organizations that are pursuing technology designed to preserve and increase U.S. competitiveness in software engineering. The symposium provides a valuable opportunity to our customers and ourselves, not only to learn about mutually beneficial activities but also to interact, to learn of emerging software engineering technology applications, and to provide feedback to one another.

Since the inception of the symposium in 1987, attendance at the event has increased from 360 to 1,200. More importantly, an increasing number of presentations are given by our customers.

Many other events are held for customers who are interested in specific aspects of software engineering or SEI work; for example, the Risk Conference, the Conference on Software Engineering Education, and Visitors Day. These and other events are in our product lists and are described in various sections of this document.

4.5 Education in Technology Transition

Education plays a significant role in technology transition. Our charter recognizes its importance as a mechanism for improving software engineering practice. The charter states:

"The SEI shall develop and conduct courses and seminars with respect to the evolving state of the art and practice in software-intensive systems. It shall also influence software engineering curricula development throughout the education community."

We believe that (1) properly educated people are better prepared to engineer software for increasingly complex systems, and that (2) the education community has an important role in establishing software engineering as a recognized discipline and profession.

Three types of education are important to the development of a highly qualified software engineer. The first is the initial education that prepares the engineer for entry into the profession; this is normally provided by the academic community in the form of bachelor's and master's degree programs. The second is continuing education that enables the engineer to stay current in a rapidly evolving discipline; this is provided through a variety of mechanisms, including in-house education programs, education vendors, university courses, and professional conferences and publications. The third type of education enables an engineer or an organization to adopt and use specific new technologies; that education and associated training are normally provided in-house or by the technology vendor.

Currently, there are no undergraduate and only a few graduate programs in software engineering in U.S. universities, so software engineers continue to enter the profession with inadequate education. This increases the burden on continuing education, which must address deficiencies in preparation as well as advances in the discipline. Very large software organizations may have an internal continuing education capability, but the majority of software engineers have few opportunities for continuing education.

Our vision is a software engineering community in which all types of education are available and of high quality. In addition, we envision software engineering to be an accepted academic and professional discipline. To achieve our vision, SEI education activities seek to provide or influence all three types of software engineering education. In keeping with the charter, we focus our activities in two areas: developing and delivering courses and associated courseware, and influencing curriculum development. Each is elaborated below.

4.5.1 Course Development and Delivery

We develop course offerings and accompanying courseware that address the state of the art and practice in both the management and technical aspects of software engineering. Development of these products is a collaborative effort involving instructional designers, graphics and video producers, communication and editing experts, and software professionals. When

the SEI has a technology that is mature enough for transition to the software community through education, SEI courseware reflects this expertise. Delivery mechanisms for courses include live offerings, transition through third-party educators, and transmission over satellite.

We also provide guidance and advice to educators and trainers. These services manifest themselves in instructor training offerings, conferences focused on the needs of software engineering educators, consultation with instructional designers, presentations and panels at professional society events, and dissemination of information about software engineering education consortia.

The end users for our education products and services are professionals in industry and government who acquire, develop, and maintain software. We reach our end users through people and institutions in the existing educational infrastructure of the United States (the higher education system, education vendors, in-house educators, government schools, and satellite broadcasters such as National Technological University).

4.5.1.1 Five-Year Plan

4.5.1.1.1 Goals

Our goals are (1) to assure that by 1999 high-quality software engineering education is widely available through traditional channels and existing infrastructure (in-house programs, university programs, vendors, and national satellite broadcasting), with effective use of education delivery technologies, and (2) to raise by 1999 the accepted educational standard for practicing software engineers and their managers.

The achievement of these goals will be demonstrated by the existence by 1999 of a well-populated national network of software engineering education delivery. The network will be composed of (1) universities offering graduate and continuing education programs in software engineering, (2) satellite receiving sites at 80 percent of major industrial and government sites, where high-quality software engineering courses are being delivered remotely and where interactive executive seminars involve managers in better decision making for software production and maintenance, (3) in-house programs at 80 percent of major industrial and government sites where software development and maintenance are key activities, and (4) geographically dispersed consortia of university and industry members (often associated with SPIN groups) who pool resources and expertise to satisfy their software engineering educational needs.

This increased availability and exposure to the content and delivery quality of existing software engineering courseware will lead to higher expectations for quality offerings and for well-qualified software engineers. By 1999 software engineers and their managers will routinely engage in software engineering education because it is recognized as essential to maintaining national competitiveness in the field and because it is available at a reasonable cost and with

minimal disruption to personal and business life. Software development organizations will show this increased focus on employee development through corporate or site software-specific training plans, employee recognition for educational achievement in software engineering, and the inclusion of educational activities in each employee's development plan.

4.5.1.1.2 Strategy

The strategy for accomplishing our goal is the following:

- Develop and deliver courses, courseware, and seminars to both universities and continuing education organizations. Maximize distribution through the use of satellite broadcasting where appropriate.
- Develop and deliver software engineering courses appropriate for academic or industry/government environments, with videotaped lectures and accompanying support materials, such as sample examinations and completed exercises.
- Conduct courses for executives to help organizations move to higher levels on the SEI CMM. Topics include process improvement, quality improvement, productivity improvement, risk management, and metrics. These courses are designed to encourage leaders and decision makers to develop and implement improvement plans for their organizations.
- Use a team approach to course development so that economies of scale and reuse can be maximized, with tailoring of products for specific audiences (academic, non-academic) and delivery mechanisms.
- Train instructors in industry and government to use SEI courseware in their programs.
- License SEI video-based courseware to education vendors to maximize dissemination to the software engineering community. Education vendors have strong existing marketing and delivery capability and reach large segments of the population, most of which do not have established in-house education groups who might use SEI courseware on their own.

4.5.1.1.3 Potential Products

Directory of industry and university consortia (1Q 1994). This directory will list industry and university consortia that focus on software engineering education. It will identify the geographical location of each consortium, its purpose, points of contact, and any relevant organizational information and history. The directory can be used by SEI customers interested in finding a consortium that will help satisfy their software engineering education needs.

Course on real-time software design and development (1Q 1994). This is an academic course that can be used by educators in colleges and universities to teach software design principles and software construction techniques that are particularly applicable to real-time systems. The course has a major emphasis on practical application for which Ada-based technology forms the core of the working environment.

Revision of academic course Software Construction with Ada (2Q 1994). This is a currently available course that can be used by educators in colleges and universities. The course focuses on large-scale software and its development using software components technology and the Ada language. The updated version will include Ada 9X revisions.

Update of academic course Software Design (2Q 1994). One of the early courses created by the SEI for college instructors, Software Design introduces several different methods and languages for expressing designs, and addresses issues of how to make design decisions and evaluate the designs of others.

Update of academic course Software Verification and Validation (4Q 1994). One of the first SEI courses for educators, Software Verification and Validation addresses the theory and practice of ensuring high-quality software products. Topics include quality assessment, proof of correctness, testing, and limitations of verification and validation methods.

Continuing education courses (4Q 1994). Two to three courses that support key process areas at levels 2 and 3 of the SEI CMM will be developed. Candidate topics include software product engineering and issues related to organizational process focus. The courses are ultimately intended for practitioners; the SEI offerings include a train-the-trainer component that prepares industry and government instructors to teach the course content in their own organizations.

Faculty development seminars (1994-1998). These seminars are held in conjunction with education-oriented conferences such as the ACM SIGCSE (Association of Computing Machinery, Special Interest Group for Computer Science Education) Technical Symposium. The purpose of these seminars is to help university educators improve their ability to teach software engineering topics. Topics for the coming years will depend upon the results of the ongoing curriculum work described in Section 4.6.2.

Two courses relating to reuse (1995). These will be continuing education courses, one focusing on management issues and one addressing more technical topics. Proposed titles are Managing Software Reuse and Software Reuse Technology.

Continuing education courses (1995-1998). We will focus development efforts on software engineering courses that support CMM key process areas, though we will address special topics that may arise in the future. One to two courses will be developed each year, with each offered twice a year. Topics may include the development of training programs and integrated software management.

4.5.1.2 One-Year Plan

4.5.1.2.1 Context

This plan continues the work described in the previous year's plan. In 1993, we completed two new academic courses, one new practitioner course, and two executive courses. We also executed our plans for updating older courses and for delivering courses. Through a relationship with NTU, SEI academic courses were broadcast by satellite for the first time. In 1994, product development, upgrading, and delivery will continue with an emphasis on more synergism between academic and non-academic development activities. In addition, we will build on the relationship we formed with NTU, expanding our course delivery through this transition partner.

4.5.1.2.2 Core Funded Activities

Develop two or three courses for practitioners that support the key process areas at levels 2 and 3 of the SEI CMM. Selection of courses will be made in late 1993; possible topics include software product engineering and issues related to organizational process focus.

Maintain currently offered practitioner courses, train-the-trainer materials, and executive courses as needed to sustain high quality of offerings and keep the content up to date.

Maintain up-to-date versions of SEI-recommended core courses for academic degree programs in software engineering.

Expand delivery mechanisms for courses through the use of third parties such as NTU and education vendors. More specifically, we will deliver all the academic courses via NTU, either as direct delivery or through another university acting as our agent (such as we do now with North Carolina State University). These will be available to more than 300 sites including dozens of U.S. military installations. We will also present one or more executive courses through NTU, and will obtain three to five licensing agreements with education vendors.

Develop and maintain a directory of industry/university consortia focusing on the development and delivery of software engineering education to members.

Deliver these courses at an SEI location (twice each):

- Instructor Training in Software Requirements Engineering
- Instructor Training in Software Design
- Software: Profit through Process Improvement
- Software Quality Improvement
- Software Productivity Improvement
- Software Risk Management
- Managing Software Development with Metrics

Deliver the above courses or tutorials based upon the courses at client sites, conferences, or education consortia events, as negotiated.

Present at least three papers/panels on effective course delivery at appropriate national conferences.

4.5.1.2.3 TO&P Funded Activities

No TO&P-funded activities are planned for 1994.

4.5.2 Curriculum Influence

Although SEI course development and delivery activities tend to influence other education providers, there are opportunities for more direct influence on academic software engineering education. U.S. universities have the potential to produce ten to twenty thousand new software engineers per year, and to provide continuing education to thousands more. SEI activities are designed to help the academic community realize that potential.

4.5.2.1 Five-Year Plan

4.5.2.1.1 Goals

The SEI works to increase the amount of software engineering taught in U.S. colleges and universities, thus preparing well-qualified students for the software engineering community in industry and government. We will measure our success by the number of software engineering programs and courses that come into existence.

Our specific goals are to:

- Continue the growth of master's programs in software engineering, at a rate that will double the number by 1999.
- Establish the first several (three to five by 1999) university programs leading to a degree of bachelor of science in software engineering (BSSE).
- Triple the number of software engineering courses being taught in U.S. colleges and universities.

4.5.2.1.2 Strategy

There are many barriers that prevent universities from creating software engineering degree programs. Among the more important are the current lack of credibility of such programs, a lack of detailed model curricula, the relative lack of textbooks and other educational materials to support teaching software engineering, the lack of qualified faculty, the lack of financial and other resources, and the perception that industry is not demanding software engineering graduates. The SEI strategy to remove these barriers is to

- Develop, refine, and promote model curricula for bachelor's and master's programs in software engineering.
- Work with professional societies to establish the credibility of software engineering curricula.
- Work with industry and government software organizations to identify and publicize their needs for software engineers.
- Develop and disseminate educational materials that facilitate the teaching of software engineering.
- Help university faculty develop their abilities to teach and conduct research in software engineering.
- Offer direct advice to universities on curriculum design and implementation.

4.5.2.1.3 Potential Products

Conference on Software Engineering Education (held annually). Held in cooperation with ACM and the IEEE Computer Society, this conference gives educators and others the opportunity to share experiences and expand their knowledge of software engineering education and training. The conferences includes in-depth tutorials, formal presentations, exhibits, and many opportunities for informal discussion.

1994 SEI Report on Software Engineering Education (4Q 1994). This report is primarily for university educators, but readers also include members of the industry and government community which university students will enter upon graduation. The report will discuss the content and structure of software engineering curriculum and issues such as accreditation, faculty development, and the evolution of software engineering from (or within) computer science.

Educational materials (1994-1998). These materials support educators (primarily those in universities) in teaching software engineering topics. Educational materials may be lecture notes, exercises, examples, case studies, or annotated bibliographies. Some are multimedia packages that include diskettes or videotape. All provide pedagogical advice on how to use the package. In 1994, plans are to publish 12 sets of educational materials, one set a month. Potential topics are software process, formal methods, requirements specification, object-oriented methods, and software architectures. Future tapes depend on the results of ongoing curriculum design work.

Revised model curricula (1995-1998). We will revise and publish updated versions of curricula for bachelor's and master's programs in software engineering.

4.5.2.2 One-Year Plan

4.5.2.2.1 Context

This plan continues the work described in previous plans. Now that we have first versions of model curricula for graduate and undergraduate software engineering programs, we will increase the emphasis in 1994 on creating educational materials and on working directly with universities who want to implement those curricula. There will also be increased emphasis on achieving more widespread awareness of SEI curriculum work and widespread distribution of products that support software engineering educators.

4.5.2.2.2 Core Funded Activities

Approximately half the effort to influence curriculum development will be devoted to creating educational materials. The other half of the effort will include all other product development and strategic activities. This includes designing curriculum; writing technical reports; working with universities, industry, and professional societies; conducting the Conference on Software Engineering Education (CSEE); and disseminating new and existing products.

The products resulting from our core-funded activities are described in Section 4.5.2.1.3, Potential Products.

4.5.2.2.3 TO&P Funded Activities

No TO&P-funded activities are planned for 1994.

4.6 Services in Technology Transition

Responding to software engineering managers' and practitioners' needs for advice and guidance on software engineering improvements, the SEI develops, delivers, and transitions services that help SEI customers improve their ability to define, develop, maintain, and operate software-intensive systems. To accelerate the widespread adoption of effective software practices, we work with organizations that are influential leaders in the software community, promote the development of infrastructures that support the adoption of improved practices, and transition capabilities to government and commercial partners for use with their customer organizations.

Our role is to provide direct support and guidance to DoD and other government organizations. Our aim is to help them build their internal capacity to initiate and sustain continuous improvement in their software development and maintenance process and in the processes they use to adopt and institutionalize new technologies. Our experience tells us that routine and effective transition of technology occurs more effectively once an organization has defined its software management processes and occurs more profitably when those processes are mature.

Since an organization must first learn to manage its software processes before profiting from the adoption of new technologies, our first focus is helping client organizations build and sustain continuous software process improvement. This is the foundation for continuous technology transition. Once an organization becomes effective at managing its technical process, it can apply its planning, organizing, and management skills to the adoption of other (non-process) technologies that apply to the organization's software development activities.

As an example of the type of organizations with which we work closely, these organizations are four of our major TO&P customers for SEI services as of June 1, 1993: Army Materiel Command, Air Force Materiel Command, Air Force Standard Systems Center, and ARPA's STARS program.

4.6.1 Five-Year Plan

4.6.1.1 Goals

It is our goal to foster improvement in software practice by working directly with the DoD software community and assisting the community's efforts to institutionalize continuous process improvement and technology transition. It is our plan to support our customers' improvement efforts so that by 1999 SEI process improvement customers will be operating with basic software management processes in place and many will be using a standardized, integrated, organization-wide software process. Because of this, they will have in place standard planning and organizing processes for improvement that they can apply to technology transition. Many

of these organizations will have developed the capacity for long-term continuous improvement and will have organizational infrastructures in place to facilitate both improvement and continuous technology transition.

By 1999, software organizations will have acquired more experience in managing the continuous changes associated with process improvement and technology transition, and will have in place strategies for managing continuous change more effectively than they do in 1993. These organizations will have in place units that are focusing on technology transition in support of their organization's ongoing improvement activities and the ability to meet the needs of rapidly changing market and technological environments.

The five-year plan of services in technology transition is to provide direct support to organizations engaged in continuous process improvement; help those that are ready to retarget some of their efforts toward continuous technology transition; provide a structured approach to technology transition; and transfer that approach to DoD software organizations. We also plan to develop and transfer approaches, tools, and techniques that support the transition of technologies related to technical environments, managerial process environments, and organizational environments.

4.6.1.2 Strategy

The strategy of services in technology transition is to:

- Continue developing and providing a structured approach to continuous process improvement.
- Provide direct support to software organizations that will enable them to develop their internal capacity for (1) improving their software processes, (2) managing the organizational and personnel changes associated with process improvement, and (3) creating an infrastructure and strategies for continuous technology transition.
- Work directly with software organizations to prepare them for effective technology transition, and advise them on interim technologies that can be used in support of continuous technical and nontechnical process improvements.
- Provide training and assistance to support continuous technology transition.
- Transfer to the DoD software community the capability to deliver the training and services we currently provide.
- Continue developing approaches and tools for enabling organizations to undertake technology transition more effectively.

4.6.2 One-Year Plan

4.6.2.1 Context

During 1993 the SEI has worked with its customers to develop, pilot, and refine the basic structures and mechanisms necessary to support process improvement, technology transition, and organizational development activities. Initial work on technology transition strategies and guidelines, process improvement strategies and guidelines, and basic material for technical and nontechnical organizational interventions has been completed. In addition, work with clients has refined the SEI approach to the transition of technical assessments and risk assessments. These efforts have laid the foundation for broader transition activities during 1994.

4.6.2.2 Core Funded Activities

No core-funded activities are planned for 1994.

4.6.2.3 TO&P Funded Activities

During 1994, the SEI will continue to concentrate on providing services to the DoD software community. We plan to do the following:

- Advance the risk technology and method of enabling clients to identify, manage, and communicate software risks in a systematic and structured way; integrate this strategy and method with other process improvement activities.
- Pilot the integration of open systems architectures, CASE, reuse, and architecture models into ongoing client improvement programs.
- Continue the strategy of transferring the capability to teach the currently existing Managing Technological Change course and the Consulting Skills Workshop to DoD organizations. Conduct trainer certification programs for these activities.
- Provide organizational services to clients that enable them to conduct software process improvement or technology transition activities effectively. Work with senior management in executive team development, vision setting, strategic planning, and aligning organizational systems to support innovation. Further, we will work with client organizations to build sustainable infrastructures for improvement, analyze nontechnological barriers to improvement or technology transition, and provide advice and guidance on organizational issues affecting clients' ability to meet their strategic goals.

- Work with client organizations to refine and document approaches to planning, organizing, and managing continuous improvement efforts. The resulting improvement roadmaps will be piloted in both DoD and commercial organizations. Use this activity to integrate our evolving approaches to process improvement and provide feedback into new product and service development activity.
- Broaden the DoD and software community process improvement and technology transition infrastructure by expanding the SEI network of distribution partners; licensing these partners to distribute additional SEI products, services, and technologies; and sponsoring public forums such as the annual SEPG National Meeting and facilitating semi-annual process improvement and technology transition workshops.

5 Relationship of Focus Areas to Software Topics

The SEI has grouped its particular technical activities in the four focus areas of process, risk, methods and tools, and real-time systems to address root causes of software problems. These focus areas continue to be appropriate groupings of the technical activities for the SEI. The same root causes, however, can be seen with equal validity from other perspectives. Current perspectives of interest include reuse, reengineering, testing, software maintenance, simulation, and open systems.

In the subsequent paragraphs, each of these topics of current interest will be discussed in the context of the SEI activities to which they relate.

5.1 Reuse

Software reuse involves the application of existing solutions to the problems of system development. In its methods and tools focus area, the SEI has synthesized and expanded on several traditional uses of software reuse and reengineering within the context of domain analysis and structural modeling (see Section 3.4.1.2).

Although most organizations do some kind of reuse in an unordered manner, the cost effectiveness of reuse grows when the assets are collected into a library and given parameters. The SEI is collaborating with other DoD initiatives in this area, such as Corporate Information Management (CIM), STARS, and CARDS.

In addition, the SEI is developing the basis for an expansion of the reuse model by focusing on a more comprehensive domain analysis. The underlying theory and knowledge of domain experts and existing software is currently being collected, organized, and analyzed within a specific area of knowledge and experience. The SEI has developed a feature-oriented domain analysis (FODA) method that develops a domain model and one or more software architectures to support the model (see SEI technical reports CMU/SEI-90-TR-21 and CMU/SEI-91-TR-28). One architectural method is referred to as structural modeling. These methods have been applied in the movement control domain (Communications-Electronics Command, Army Tactical Command and Control System), flight simulation (Ada Simulator Validation Program (ASVP), Advanced Millimeter-Wave Seeker, Aeronautical Systems Center Training Simulator Office), design simulation (Air Force Electronic Combat Office), and radar (construction of a radar to operationally simulate signals believed to originate within the Soviet Union (CROSSBOW-S)), Joint Modeling and Simulation System (J-MASS).

The domain analysis concept can be used to give added value to other complementary approaches. For example, a reverse engineering approach establishes a model of existing software for potential future forward engineering. As a result, a model established from a reverse engineered system can be one important component of an overall domain model for an area.

5.2 Reengineering

In the last several years, there has been significant discussion about the legacy of software systems and reengineering. As a result of this attention, reengineering tools have begun to appear. Their focus is primarily on deriving information from code of legacy systems (reverse engineering), on restructuring and retargeting code, and on mapping derived design information into a new implementation. In particular, several tools exist to migrate information systems implemented in COBOL to new platforms and to upgrade their data representation into a relational form.

While reengineering tools will help in certain aspects of reengineering, reengineering is no more about tools than engineering is about tools. Just as engineering implies a disciplined process supported by engineering methods and automated tools, reengineering practice requires a disciplined process supported by methods and tools. In short, reengineering is viewed as an engineering problem that requires a quantitative analysis of the problem, and consideration of engineering tradeoffs in its solution.

In response to ARPA guidance, the SEI proposes to address reengineering issues through its work in methods and tools (see Section 3.4.1.2) as well as contributions from throughout the SEI. Activities specific to reengineering are the development of a conceptual framework for reengineering, the assessment of the state of current practice in reengineering, and an SEI-sponsored workshop on reengineering as a forum in which the foundation for a guide to best reengineering practice will be laid (see Sections 3.4.1.3 and 3.4.2.2). It is expected that the SEI will be involved in external projects that have a reengineering character as case studies, including the STARS demonstration projects. The SEI also expects to cooperate with existing activities related to reengineering, in particular the Joint Logistics Commanders (JLC) Joint Policy Coordinating Group on Computer Management addressing reengineering policy making. More details are described in the SEI report *Reengineering: An Engineering Problem* (CMU/SEI-93-SR-5).

5.3 Testing

Due to budget cuts, the feasibility study on software testing that we had planned for 1993 was never started. This study would have investigated current and emerging technology for software testing, thus complementing our work on fault tolerance technology.

During 1994, if funding and staffing permit, we will attempt to address some of the goals of the software testing study. Anticipated results could include a standard vocabulary for describing system testing requirements, and assessments of available testing methodology that addresses design for tests, criteria for determining the adequacy of tests, and the relationship of testing to software verification and certification. The best practices in software testing could then be incorporated in the Software Developer's Handbook.

5.4 Software Maintenance

The SEI does not distinguish between maintenance and development since we believe in a continuous evolutionary model of software in which "maintenance" activities are additional iterations of the spiral. Maintenance-related issues are included in the context of reengineering (see above and Sections 3.4.1.2, 3.4.2.2, and 3.4.2.3) and in the work on software process, risk, methods and tools, and real-time distributed systems.

5.5 Simulation

Our technical approach to simulation is based on structural modeling, a domain-specific software architecture design methodology. Structural modeling was developed to support real-time training simulators and has been successfully used on B-2, C-17, Special Operations Forces Aircrew Training System (SOF-ATS), and Simulator for Electronic Combat Training (SECT) programs. In addition, the *Introduction to Structural Modeling* is currently part of the bidder's library of the Air Force Aeronautical Systems Center Training Simulator Office (ASC/YT), and it is cited in requests for proposals.

Structural modeling is expanding to additional domains. Most of these activities are described under the subsection on structural modeling technology products in Section 3.5.1.3. Additional SEI work on simulation includes a recent study of languages for simulation (sponsored by the Defense Modeling and Simulation Office) and development of automatic benchmarks for distributed simulation (sponsored by Simulation, Training, and Instrumentation Command (STRICOM)) to be reported in a 1994 report: *Benchmarking Technology for Distributed Interactive Simulation (DIS)*.

5.6 Open Systems

The SEI contributes to the development of open systems in several ways. Our focus on process improves the openness of process-centered, computer-aided software engineering (CASE) environments by promoting a more defined and standardized software process (process asset library, capability maturity model, and the International Standards Organization (ISO) 9000). Our focus on technical risk assessment addresses issues regarding the openness of systems and the resulting tradeoffs. Through our focus on methods and tools, we address development of application systems through the use of domain models and common software architectures leading toward more flexible system implementations that can be adapted and reused. One of the domains is CASE environments (i.e., the integration of CASE tools into an effective software engineering environment), and the SEI is involved in several environment standardization efforts. Through our focus on real-time distributed systems, we are addressing issues in applications—that is, elements of an application implementation relevant to the execution of the application system.

Through our focus on real-time distributed systems, we contribute to the use of open architectures to promote reuse and interoperability. We participate in the development of the IEEE 1003 family of standards (POSIX) with special emphasis on meeting dual-use requirements. We are also examining issues associated with use of open architectures, such as the metrics that are necessary for system developers. These activities are described in Section 3.5.1.3.

Appendix A Currently Available SEI Products and Services

| PROCESS | Product | Manager | Practitioner | Educator |
|---------|--|---------|--------------|----------|
| | Software Capability Evaluation Overview Seminar (course) | ▼ | ▼ | ▼ |
| | Software Capability Evaluation Team Training (course) | ▼ | ▼ | |

| RISK | Product | Manager | Practitioner | Educator |
|------|---|---------|--------------|----------|
| | Software Risk Management Conference (event) | ▼ | ▼ | |

| METHODS AND TOOLS | Product | Manager | Practitioner | Educator |
|-------------------------|--|---------|--------------|----------|
| | Past, Present, and Future of Configuration Management Automation (videotape) | ▼ | ▼ | ▼ |

| REAL-TIME SYSTEMS | Product | Manager | Practitioner | Educator |
|----------------------|---|---------|--------------|----------|
| | Hartstone (software) | | ▼ | |
| | Introduction to Rate Monotonic Analysis (videotape) | | ▼ | ▼ |
| | Serpent (software) | | ▼ | |

| EDUCATION IN TECHNOLOGY TRANSITION | Product | Manager | Practitioner | Educator |
|------------------------------------|--|---------|--------------|----------|
| | Applying Software Engineering Skills to Writing (videotape) | | ▼ | ▼ |
| | Conference on Software Engineering Education (event) | | | ▼ |
| | Executive Leadership for Software (videotape) | ▼ | | ▼ |
| | Formal Methods in Software Engineering (course) | | | ▼ |
| | Software and Some Lessons from Engineering (videotape) | | ▼ | ▼ |
| | Software Construction with Ada (course) | | | ▼ |
| | Software Design, Creation, and Maintenance (course) | | | ▼ |
| | Software Productivity Improvement (course) | ▼ | | |
| | Software: Profit Through Process Improvement (course) | ▼ | | |
| | Software Project Management for Academia (course) | | | ▼ |
| | Software Project Management for Industry and Government (course) | | | ▼ |
| | Software Quality Improvement (course) | ▼ | | |
| | Software Requirements Engineering (course) | | | ▼ |
| | Software Risk Management (course) | ▼ | | |
| | Software Specification (course) | | | ▼ |
| | Software Verification and Validation (course) | | | ▼ |

| SERVICES IN TECHNOLOGY TRANSITION | Product | Manager | Practitioner | Educator |
|---|---|---------|--------------|----------|
| | Consulting Skills Workshop (course) | ▼ | ▼ | |
| | Executive Overview for Managing Technological Change (course) | ▼ | | |
| | Managing Technological Change (course) | ▼ | ▼ | ▼ |
| | SEPG National Meeting (event) | ▼ | ▼ | |

| SEI-WIDE | Product | Manager | Practitioner | Educator |
|----------|-----------------------|---------|--------------|----------|
| | SEI Symposium (event) | ▼ | ▼ | ▼ |
| | Visitors Day (event) | ▼ | ▼ | ▼ |

Appendix B Resident Affiliates

| | Name | Total to Date | One Affiliate in Residence as of 19 July 1993 |
|-----------------|------------------------------------|---------------|---|
| INDUSTRY | AT&T Bell Labs | 1 | |
| | Boeing | 1 | |
| | Computer Sciences Corporation | 4 | ◆ |
| | GE Aerospace | 2 | |
| | General Dynamics | 1 | |
| | GTE Government Systems | 3 | ◆ |
| | Hughes Aircraft Company | 4 | |
| | IBM | 5 | ◆ |
| | Lockheed Missiles & Space Co., Inc | 1 | |
| | Pacific Bell | 2 | ◆ |
| | Paramax | 2 | |
| | Process, Inc. | 1 | ◆ |
| | Raytheon Company | 1 | |
| | Siemens Corporate Research | 1 | ◆ |
| | SYSCON Corporation | 1 | |
| | TeleSoft | 1 | |
| | Texas Instruments | 2 | ◆ |
| | Westinghouse Electric Corporation | 3 | |

| | | Name | Total to Date | One Affiliate in Residence as of 19 July 1993 |
|-------------------|------------------|--|---------------|---|
| GOVERNMENT | Navy | Naval Air Development Center | 2 | |
| | | Coastal Systems Station | 1 | |
| | | Naval Ocean Systems Center | 3 | ❖ |
| | | Naval Surface Warfare Center | 3 | |
| | | Naval Undersea Warfare Engineering Station | 1 | |
| | | Naval Undersea Weapons Engineering Station | 1 | |
| | | Naval Weapon Center | 2 | |
| | | Naval Air Warfare Center | 1 | ❖ |
| | Army | Communications-Electronics Command | 6 | |
| | Air Force | Air Combat Command | 1 | |
| | | Air Force Institute of Technology | 5 | |
| | | Air Logistics Center | 1 | |
| | | Electronic Systems Center | 3 | ❖ |
| | | Space Command | 1 | |
| | | Standard Systems Center | 1 | |
| | Other | Department of Defense | 6 | ❖ |

Appendix C Transition Partners

Below are current SEI transition partners who make products available either separately from, or in addition to, direct distribution from the SEI.

| Company | Product |
|--|--|
| <p>Telos Consulting Services Gentry Gardner 5526 N. Academy Blvd., Suite 203 Colorado Springs, CO 80918 Phone: (719) 260-1333 FAX: (719) 260-0022 E-mail: ggg@telos.com</p> | <p>RMA-based course: • Guaranteeing Real-Time Performance</p> |
| <p>Kluwer Academic Publishers 101 Philip Drive Norwell, MA 02061 Phone: (617) 871-6300 FAX: (617) 871-6528 E-mail: kluwer@world.std.com</p> | <p>RMA Handbook (<i>A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems</i>)</p> |
| <p>National Technological University Ellen Stafford 700 Centre Avenue Fort Collins, CO 80526-1842 Phone: (303) 495-6424 FAX: (303) 485-0668 E-mail: ellen.stafford@ntupub.ntu.edu</p> | <p>SEI Academic and Continuing Education Courses</p> |
| <p>Tri-Pacific Consulting Corporation Russell Plain 1070 Marina Village Parkway, Suite 202 Alameda, CA 94501 Phone: (510) 814-1770 (800) 438-8064 FAX: (510) 814-1788 E-mail: 76434.117@CompuServe.COM</p> | <p>RMA-based courses: • RMA Overview • Software Design Using RMA • Systems Engineering Using RMA • RMA Management Overview</p> |

| Company | Product |
|--|------------------------------------|
| Defense Technical Information Center ATTN: FDRA Cameron Station Alexandria, VA 22304-6145 Phone: (703) 274-7633 | SEI publications |
| Research Access Inc. 800 Vinial Street Pittsburgh, PA 15212 Phone: (412) 321-2992 or 1 (800) 685-6510 FAX: (412) 321-2994 | SEI publications |
| National Technical Information Service Department of Commerce Springfield, VA 22161-2103 Phone: (703) 487-4600 | SEI publications |
| Addison-Wesley Computer Science and Engineering Marketing Jacob Way Reading, MA 01867 Phone: (617) 944-3700, x2396 | SEI Series in Software Engineering |
| MIT X Consortium Distribution Library | Serpent |
| American Management Systems Inc. Ron Brown 1455 Frazee Road, Suite 315 San Diego, CA 92108 Phone: (619) 297-5800 FAX: (619) 297-3005 | SPA |
| Arthur D. Little Inc. Joseph Puffer (617) 498-5691 Paul Scheib (617) 498-5692 Acorn Park Cambridge, MA 02140-2390 FAX: (617) 498-7262 E-mail: /PN=PAULA.M.BENNETT/0=ADLITTLE/ ADMD=TELEMAIL/C=USS/@sprint.com | SPA |

| Company | Product |
|--|----------------------------------|
| Booz-Allen & Hamilton Joel Stream 4330 East West Highway Bethesda, MD 20814-4455 Phone: (301) 951-4698 FAX: (301) 907-4481 | SPA |
| Dayton Aerospace Associates Inc. Jim Waite 4141 Colonel Glenn Highway Suite 252 Beavercreek, OH 45431 Phone: (513) 426-4300 FAX: (513) 426-1352 E-mail: 73052,1026@compuserve.com | SPA |
| Defense Information Systems Agency Evelyn M. DePalma Center for Information Management 701 South Courthouse Road Arlington, VA 22204-2199 Phone: (703) 285-6584 FAX: (703) 285-6594 | SPA (government clients only) |
| Digital Equipment Corporation Mark Rabideau 711 Nob Hill Trail Franktown, CO 80112 Phone: (303) 649-3484 FAX: (303) 660-9217 | SPA |
| pragma Systems Corporation Rebecca Bowerman 8704 Lee Highway Suite 303 Fairfax, VA 22031 Phone: (703) 560-4669 FAX: (703) 849-8839 E-mail: pragma@grebyn.com | SPA |

| Company | Product |
|---|---|
| Process Inc. Louise Hawthorne 829 Norwest Road Suite 806 Kingston, Ontario K7P 2N3 Phone: (613) 531-9972 FAX: (613) 384-9383 E-mail: lhawthor@sei.cmu.edu | SPA |
| Process Enhancement Partners Inc. Judah Mogilensky (301) 589-1037 Mark Manduke (703) 222-6159 1902 Rookwood Road Silver Spring, MD 20910 FAX: (301) 589-0524 E-mail: jmogilen@sei.cmu.edu (Judah Mogilensky) mmanduke@sei.cmu.edu (Mark Manduke) Compuserve: 72172,1012 | SPA |
| Software Productivity Consortium Jerry Decker (703) 742-7216 Jack Hofmann (703) 742-7279 SPC Building 2214 Rock Hill Road Herndon, VA 22070 FAX: (703) 742-7360 E-mail: decker@software.org (Jerry Decker) hofmann@software.org (Jack Hofmann) | SPA Executive Leadership for Software videotape (available to SPC members) |
| Technology Applications Inc. Dale Luddeke 6101 Stevenson Avenue Alexandria, VA 22304 Phone: (703) 461-2116 FAX: (703) 461-2299 | SPA |

| Company | Product |
|---|--|
| <p>Technology Integration Center Maj. Mark Lewis U.S. Air Force HQ AFC4A/XPSP 203 West Losey Street, Room 1020 Scott AFB, IL 62225-5219 Phone: (618) 256-5697 FAX: (618) 256-2874 E-mail: lewis@afc4a.safb.af.mil (Maj. M. Lewis)</p> | <p>SPA (government clients only)</p> |

References

- [Conner 82] Conner, D. R.; and Patterson, R. W. "Building Commitment to Organizational Change." *Training and Development Journal*, April 1982.
- [Dion 92] Dion, R. "Elements of a Process-Improvement Program (Software Quality)." *IEEE Software* 9, 4 (July 1992): 83-85.
- [Grady 89] Grady, R. B., and Caswell, D. L. *Software Metrics: Establishing a Company-Wide Program*. Englewood Cliffs, N.J.: Prentice-Hall, 1989.
- [Humphrey 91] Humphrey, W. S.; Snyder, T. R.; and Willis, R. R. "Software Process Improvement at Hughes Aircraft." *IEEE Software* 8, 4 (July 1991): 11-23.
- [Lipke 92] Lipke, W. H.; and Butler, K. L. "Software Process Improvement: A Success Story." *CrossTalk*, November 1992.
- [Putnam 93] Putnam, L. H. *The Economic Value of Moving Up the SEI Scale*. Technical Report, Quantitative Software Management Inc., McLean, Va., 1993.
- [Wohlwend 93] Wohlwend, H.; and Rosenbaum, S. "Software Improvements in an International Company." *Proceedings of the 15th International Conference on Software Engineering*. Austin, Texas, May 1993.

List of Acronyms

| | |
|------------|---|
| AAS | Advanced Automation System |
| ACM SIGCSE | Association of Computing Machinery, Special Interest Group for Computer Science Education |
| AFMC | Air Force Materiel Command |
| AFMC/ESC | Air Force Materiel Command/Electronic Systems Center |
| AJPO | Ada Joint Program Office |
| ALT | Advanced Learning Technologies |
| AMC | Army Materiel Command |
| AMORE | Advanced Multimedia Organizer for Requirements Elicitation |
| ANSI | American National Standards Institute |
| ARPA | Advanced Research Projects Agency |
| ASVP | Ada Simulator Validation Program |
| ATIS | A Tools Integration Standard |
| AWACS | Airborne Warning and Control System |
| BMD | Ballistic Missile Defense |
| BSSE | bachelor of science in software engineering |
| CARDS | Central Archive for Reusable Defense Software |
| CASE | computer-aided software engineering |
| CCTT | Close Combat Tactical Trainer |
| CIM | Corporate Information Management |
| CM | configuration management |
| CMM | Capability Maturity Model |
| CMU | Carnegie Mellon University |
| COTS | commercial off-the-shelf |
| CROSSBOW-S | construction of a radar to operationally simulate signals believed to originate within the Soviet Union |
| CSEE | Conference on Software Engineering Education |

| | |
|--------|---|
| DIS | Distributed Interactive Simulation |
| DMSO | Defense Modeling and Simulation Office |
| DoD | Department of Defense |
| DSMC | Defense Systems Management College |
| DSSA | Domain Specific Software Architecture |
| DTIC | Defense Technical Information Center |
| EDRC | Engineering Design Research Center |
| ESF | Eureka Software Factory |
| ESIP | Embedded Computer Resources Support Improvement Program |
| FAA | Federal Aviation Administration |
| FFRDC | federally funded research and development center |
| FODA | feature-oriented domain analysis |
| I2M2 | intelligent, interactive multimedia |
| IBM | International Business Machines |
| I-CASE | integrated CASE |
| IEEE | Institute of Electrical and Electronic Engineers |
| IPSE | integrated project support environments |
| ISEE | integrated software engineering environment |
| ISO | International Standards Organization |
| ITC | Information Technology Center |
| JLC | Joint Logistics Commanders |
| J-MASS | Joint Modeling and Simulation System |
| KPA | key process area |
| LAS | Aircraft Software Division |
| LCSS | life-cycle software support |
| MBSE | model-based software engineering |
| MIS | management information systems |
| MTTD | mean time to defect |
| NAPI | North American PCTE Initiative |

| | |
|--------|---|
| NASA | National Aeronautics and Space Administration |
| NAVAIR | Naval Air Systems Command |
| NAVSUP | Naval Supply Systems Command |
| NAWC | Naval Air Warfare Center |
| NGCR | Next Generation Computer Resource |
| NII | National Information Infrastructure |
| NIST | National Institute of Standards and Technology |
| NOAA | National Oceanographic and Atmospheric Administration |
| NRC | Nuclear Regulatory Commission |
| NSA | National Security Agency |
| NTIS | National Technical Information Service |
| NTP | National Technology Policy |
| NTU | National Technological University |
| OC-ALC | Oklahoma Air Logistics Center |
| ONR | Office of Naval Research |
| PDM | predictive decision model |
| PDSS | post-deployment software support |
| PEO | program executive officers |
| PM | program manager |
| PSESWG | Project Support Environments Standards Working Group |
| PVM | Process Value Method |
| R&D | research and development |
| RAI | Research Access Inc. |
| RMA | rate monotonic analysis |
| ROI | return on investment |
| SAIL | Systems to Automate and Integrate Logistics |
| SCE | Software Capability Evaluation |
| SCS | School of Computer Science |
| SDIO | Strategic Defense Initiative Office |

| | |
|---------|--|
| SECT | Simulator for Electronic Combat Training |
| SEE | software engineering environments |
| SEI | Software Engineering Institute |
| SEPG | Software Engineering Process Group |
| SOF-ATS | Special Operations Forces Aircrew Training System |
| SPA | Software Process Assessment |
| SPD | Software Process Definition |
| SPICE | Software Process Improvement Capability dEtermination |
| SPIN | Software Process Improvement Network |
| SPM | Software Process Measurement |
| SRE | Software Risk Evaluation |
| SSF | Space Station Freedom |
| STARS | Software Technology for Adaptable, Reliable Systems |
| STRICOM | Army Simulation, Training, and Instrumentation Command |
| STSC | Software Technology Support Center |
| SWAP-WG | Software Action Plan Working Group |
| TBQ | Taxonomy-Based Questionnaire |
| TC | Technical Committee |
| TI | Texas Instruments |
| TO&P | technical objectives and plans |
| WG | Working Group |

REPORT DOCUMENTATION PAGE

| | | | | | |
|---|-------|--|--|---|--|
| 1a. REPORT SECURITY CLASSIFICATION Unclassified | | | 1b. RESTRICTIVE MARKINGS None | | |
| 2a. SECURITY CLASSIFICATION AUTHORITY N/A | | | 3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited | | |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A | | | | | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU/SEI-93-SR-19 | | | 5. MONITORING ORGANIZATION REPORT NUMBER(S) | | |
| 6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute | | 6b. OFFICE SYMBOL (if applicable) SEI | 7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office | | |
| 6c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213 | | | 7b. ADDRESS (city, state, and zip code) HQ ESC/ENS 5 Eglin Street Hanscom AFB, MA 01731-2116 | | |
| 8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office | | 8b. OFFICE SYMBOL (if applicable) ESC/ENS | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003 | | |
| 8c. ADDRESS (city, state, and zip code) Carnegie Mellon University Pittsburgh PA 15213 | | | 10. SOURCE OF FUNDING NOS. | | |
| | | | PROGRAM ELEMENT NO 63756E | PROJECT NO. N/A | TASK NO N/A |
| | | | WORK UNIT NO. N/A | | |
| 11. TITLE (Include Security Classification) Draft SEI Program Plans: 1994-1998 | | | | | |
| 12. PERSONAL AUTHOR(S) | | | | | |
| 13a. TYPE OF REPORT Final | | 13b. TIME COVERED FROM TO | | 14. DATE OF REPORT (year, month, day) August 1993 | |
| 15. PAGE COUNT 180 | | | | | |
| 16. SUPPLEMENTARY NOTATION | | | | | |
| 17. COSATI CODES | | | 18. SUBJECT TERMS (continue on reverse of necessary and identify by block number) | | |
| FIELD | GROUP | SUB. GR. | program plans, technical program, strategy, one-year plan, five-year plan, technology transition, products and services. | | |
| | | | | | |
| | | | | | |
| | | | | | |
| 19. ABSTRACT (continue on reverse if necessary and identify by block number) This document presents the Software Engineering Institute (SEI) strategy and one-year implementation plan for calendar year (CY) 1994, together with the SEI five-year program plan. | | | | | |
| (please turn over) | | | | | |
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input checked="" type="checkbox"/> | | | 21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution | | |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas R. Miller, Lt Col, USAF | | | 22b. TELEPHONE NUMBER (include area code) (412) 268-7631 | | 22c. OFFICE SYMBOL ESC/ENS (SEI) |